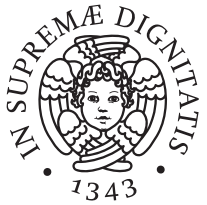

Università degli studi di Pisa
Facoltà di Economia
Facoltà di Scienze Matematiche, Fisiche e Naturali
Corso di Laurea Magistrale in Informatica per l'economia e per l'azienda
(Business Informatics)

Utilizzo di tecniche di Process Mining nel contesto del Tuscan Port Community System

Relatori:
Roberto Bruni
Giorgio Oronzo Spagnolo

Candidato:
Gabriele Bocchi

Anno accademico 2013-2014

Ai miei genitori

Abstract

Nel corso degli ultimi anni hanno assunto sempre maggiore importanza, all'interno delle aziende, la specifica ed il controllo dei processi aziendali o processi di business. Per modellare i processi di business viene utilizzata la notazione BPMN, oramai uno standard de facto nella modellazione dei processi. Questa notazione semplice ed intuitiva permette di specificare, in modo chiaro, tutte le attività che portano allo svolgimento di un processo aziendale, rendendolo facilmente comprensibile a tutti gli attori coinvolti. I processi di business permettono di descrivere sia processi intra-aziendali che extra-aziendali. Questi possono essere implementati in piattaforme più o meno automatiche che tracciano le proprie attività attraverso dei log. In questa tesi mostreremo come, utilizzando tecniche di Process Mining, è possibile estrarre conoscenza dai log, attraverso uno scenario reale costituito dalla piattaforma Tuscan Port Community System (TPCS) dell'Autorità Portuale di Livorno. In particolare, verificheremo se le azioni svolte dai vari attori, all'interno del processo di export delle merci, trovano una corrispondenza con il modello di processo associato a tale pratica.

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

INTRODUZIONE

Il mondo di oggi è caratterizzato da un ritmo di vita sempre più frenetico, da maggiori esigenze in fatto di qualità e velocità nelle realizzazione di prodotti, nel soddisfacimento delle richieste e nell'erogazione dei servizi. Per questo le aziende si trovano a fronteggiare richieste sempre più esigenti da parte degli utenti finali. Per fronteggiare questo fenomeno è aumentato lo sviluppo di nuove tecnologie che facilitano e velocizzano drasticamente queste operazioni ma al tempo stesso complicano notevolmente la parte gestionale e l'organizzazione del lavoro all'interno dell'azienda. Per questo vengono introdotte sempre nuove mansioni, inserite nuove terminologie ed è sempre più difficile comprendere perfettamente che cosa e come dobbiamo fare per svolgere al meglio il nostro lavoro.

Per affrontare tutte queste problematiche sono stati introdotti i Processi di Business. Questi si basano sull'utilizzo di modelli di processo che servono per descrivere le singole attività svolte all'interno di un processo, sia per quanto riguarda i metodi di svolgimento che la sincronizzazione, le dipendenze logiche e gli strumenti utilizzati nelle varie attività. Chiaramente questo tipo di modellazione deve avvenire in accordo con gli obiettivi dell'organizzazione che dovrà implementare il processo. Le aziende hanno sposato favorevolmente l'introduzione all'interno dell'azienda di questi modelli di processo, rendendosi subito conto che sono di immediata leggibilità, in quanto utilizzano una notazione grafica molto semplice e di facile comprensione per tutti, anche per le persone con un background limitato del contesto aziendale. Grazie alla sua semplicità infatti, i modelli di processo hanno avuto una rapida evoluzione all'interno delle organizzazioni acquistando sempre maggiore importanza. L'unico inconveniente dell'utilizzo di questi modelli riguarda l'accettazione da parte dei dipendenti di un modello strutturato e quindi più rigido rispetto alle abitudini precedenti, che lasciavano maggiore spazio alla

libera interpretazione.

L'implementazione di questi modelli all'interno di contesti reali ha portato all'introduzione di un sistema di registrazione degli eventi all'interno di file di log. Il log è un file sequenziale che viene utilizzato per registrare, in modo cronologico, tutte le operazioni che vengono eseguite. Grazie ai log è possibile ricostruire tutte le operazioni che sono state effettuate, conoscere chi e quando le ha effettuate. Questo garantisce una grande facilità nella misurazione ed un continuo monitoraggio delle attività coinvolte nel processo, permettendo così di individuare eventuali punti di dispersione, sia di tempo che di risorse, all'interno del processo. Grazie a queste misurazioni sarà possibile individuare potenziali miglioramenti al fine di rendere il processo più performante sia per l'azienda che per gli attori coinvolti.

Il controllo ed il monitoraggio di questi processi viene effettuato grazie al Process Mining, che costituisce un'area di ricerca relativamente giovane e di recente sviluppo che può essere collocata tra il Data Mining e la modellazione/analisi dei processi. Si occupa di modellare, monitorare e migliorare i processi estraendo conoscenza dai log.

Lo scopo del nostro progetto di tesi è proprio quello di mostrare l'utilizzo di queste tecniche in un contesto reale, che è quello del processo di export delle merci all'interno del porto di Livorno. L'Autorità Portuale di Livorno per andare incontro a esigenze, come la necessità di ridurre i tempi e velocizzare le procedure necessarie per lo sdoganamento della merce, ha deciso, attraverso la sottoscrizione di un protocollo d'intesa con l'Agenzia delle Dogane, di sviluppare la piattaforma TPCS (Tuscan Port Community System), che permette di velocizzare le attività da mettere in atto per esportare o importare le merci del porto. Il TPCS mette a disposizione i log di tutte le azioni svolte, all'interno della piattaforma. Utilizzando i log estratti dalla suddetta piattaforma insieme al modello di processo associato allo sdoganamento merci ed al tool ProM (un programma di Process Mining), è stato possibile verificare la correttezza delle operazioni svolte dai vari operatori, controllando così la conformità di queste ultime rispetto al modello di processo adottato.

Abbiamo raccolto molte informazioni sulle caratteristiche delle singole operazioni svolte all'interno del processo, sia in merito alla correttezza di tali operazioni che in merito alle tempistiche di svolgimento di queste ultime. Questo ha permesso di riscontrare eventuali errori commessi da parte degli attori o del sistema. Inoltre è stato possibile individuare comportamenti comuni delle istanze del processo prese in esame.

1.1 La struttura della tesi

Di seguito viene spiegato come è stata strutturata la tesi:

Nel secondo capitolo vengono descritte tutte le conoscenze di base che sono necessarie per poter comprendere il progetto di tesi. In particolare vengono descritti i Modelli di Processo, i Log, il Process Mining, il tool ProM ed il formato standard XES.

Nel terzo capitolo viene contestualizzato il progetto di tesi, descrivendo il contesto aziendale dal quale sono stati estratti i dati, soffermandosi in particolare sulla piattaforma che raccoglie questi dati (TPCS) e sul processo analizzato (processo di Export).

Nel quarto capitolo viene descritta la prima fase operativa che consiste nella preparazione dei dati ed in particolare nella trasformazione dal formato CSV al formato XES. Contestualmente viene descritto il plug-in che è stato utilizzato per eseguire questa trasformazione e le modifiche che sono state apportate su di esso, in modo da aumentarne le funzionalità.

Nel quinto capitolo vengono descritte le fasi di analisi dei log e di verifica della conformance tra il Business Process ed i log. Inoltre, vengono descritti i vari plug-in utilizzati per ogni tipo di analisi ed i risultati ottenuti.

Ed infine il sesto capitolo comprende le conclusioni che possono essere tratte da questo progetto di tesi.

Capitolo 2

BACKGROUND

In questo capitolo vengono descritte le conoscenze di base che sono necessarie per poter comprendere al meglio l'ambito in cui si è sviluppato questo lavoro di tesi.

Di seguito verranno descritti cosa sono i modelli di processo, i log ed il process mining.

2.1 Modelli di Processo

I modelli di processo sono dei modelli che descrivono le singole attività che devono essere svolte all'interno di un processo. Definire un processo significa specificare le attività, le relative procedure, le loro relazioni in termini di sincronizzazione ed in termini di dipendenze logiche, quali, ad esempio, le responsabilità delle persone coinvolte, gli strumenti che esse dovranno utilizzare, i documenti che verranno prodotti e modificati nel corso del processo. La modellazione di un processo deve avvenire in accordo con gli obiettivi dell'organizzazione che lo dovrà implementare; questo permetterà, successivamente, la sua misurazione ed il monitoraggio continuo, e consentirà l'identificazione dei punti di dispersione (di tempo e risorse) del processo e dei potenziali miglioramenti, al fine di renderlo efficace (capace di raggiungere gli obiettivi) ed efficiente (capace di sfruttare il numero minore possibile di risorse). La modellazione del processo consente, inoltre, grazie alla sua rappresentazione grafica, una rapida diffusione della conoscenza all'interno dell'Organizzazione.

Tra i modelli di processo, quelli che trovano la maggiore diffusione sono i Modelli di Business. Questa tipologia di modello si riferisce ai modelli aziendali e trova largo impiego grazie alla sua notazione facile ed intuitiva che

permette, a tutti gli attori coinvolti, di essere in grado di capire facilmente quali attività devono essere svolte e in che modo devono essere portate a termine.

I modelli BPMN

Tra le notazioni per i modelli di business la più diffusa attualmente è la **Business Process Model and Notation** (BPMN) [OMG, 2011]. BPMN è una rappresentazione grafica nata per specificare processi di business in modo facile ed intuitivo. La notazione è stata progettata per coordinare la sequenza dei processi e dei messaggi che vengono scambiati tra i diversi partecipanti al processo in una serie di attività connesse. Questo standard è stato sviluppato dalla **Business Process Management Initiative** (BPMI), che è un'organizzazione indipendente dedicata allo sviluppo di specifiche aperte per la gestione dei processi di e-Business che si estendono su più applicazioni, funzioni aziendali e partner commerciali. Questo standard viene manutenuto dalla **Object Management Group™** (OMG ™) dal 2005, anno in cui le due organizzazioni sopra citate, decisero di unire le loro attività di **Business Process Management** (BPM) per fornire uno standard di pensiero e di leadership per questo settore in crescita. Il gruppo combinato prese il nome di **Business Modeling & Integration Domain Task Force** (BMI-DTF).

Lo sviluppo di BPMN è stato un passo importante nel ridurre la frammentazione che esisteva nella grande varietà di strumenti di modellazione di processo e nella notazione che sfruttava conoscenze ed esperienze diverse. Fino alla nascita di BPMN, non vi è stata una tecnica di modellazione standard. BPMN è stato sviluppato per fornire agli utenti una reale notazione di riferimento.
La versione corrente di BPMN è la 2.0, che è in vigore dall'anno 2011.

BPMN definisce uno standard per **Business Process Diagrams** (BPD), basato su una tecnica di diagrammi di flusso su misura per i modelli grafici delle operazioni di business process.

Gli obiettivi del BPMN sono:

- fornire una notazione comprensibile da:
  - analisti che definiscono i processi;
  - sviluppatori responsabili dell'implementazione tecnologica dei processi;
  - persone che gestiranno e terranno sotto controllo i processi.

- far sì che i linguaggi nati per l'esecuzione dei processi di business (es. BPEL) possano essere visualizzati con una notazione "non tecnica".

BPMN[1] è caratterizzato da oltre 50 artefatti ma, per questo lavoro di tesi, ne è stato selezionato un piccolo sottoinsieme, senza per questo limitare l'espressività dei modelli che è possibile realizzare.

La Figura 2.1 mostra gli elementi che sono stati selezionati:

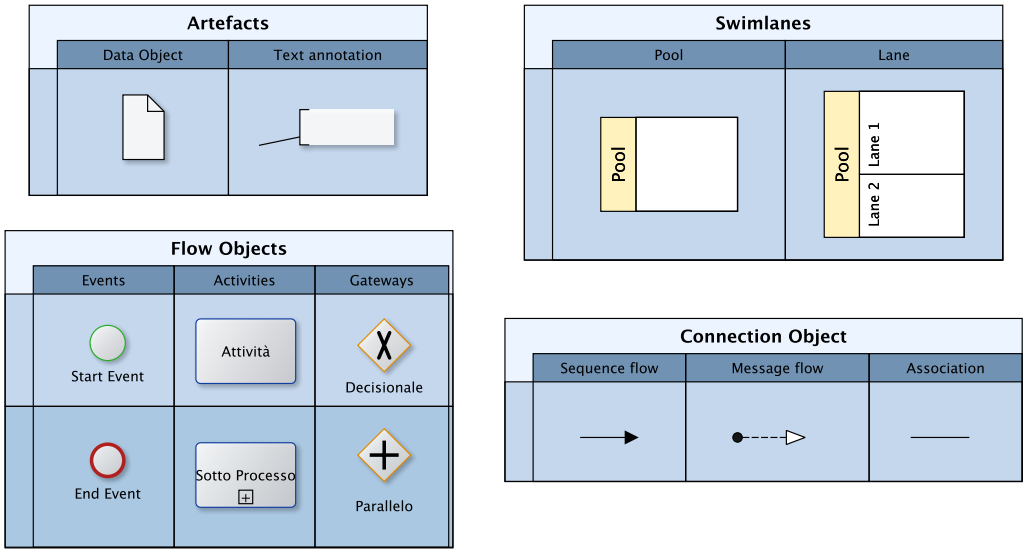

Figura 2.1: Categorie degli elementi base di BPMN

1. **Artefacts**, che si dividono in:

   - **Data object:** specifica i dati che sono richiesti o prodotti da un'attività. I data object non hanno nessun effetto diretto sulla sequenza di flusso. Vengono spesso rappresentati con l'icona del file;
   - **Text annotation:** ogni oggetto può essere associato ad una text annotation per fornire tutte le informazioni ed i documenti aggiuntivi che possono essere necessari. Una text annotation è rappresentata da una linea tratteggiata composta da punti.

2. **Swimlanes**, che si dividono in:

[1] Al seguente indirizzo è disponibile il poster dove vedere tutti gli elementi BPMN `http://www.bpmb.de/images/BPMN2_0_Poster_IT.pdf`

- **Pool:** rappresenta un partecipante (o ruolo) in un processo. È rappresentato con un rettangolo con un nome;
- **Lane:** è una sub-partizione gerarchica all'interno di un pool che viene utilizzato per organizzare e categorizzare le attività. È rappresentato da un rettangolo interno al pool che si estende per l'intera lunghezza del pool stesso.

3. **Flow objects**, che si dividono in:

   - **Event:** un evento è qualcosa che accade durante il corso di un business process. È rappresentato da un cerchio e nella nostra analisi consideriamo solo due tipologie principali:
     - **evento iniziale:** indica l'inizio di un processo, può essere opzionale, possono essercene più di uno ed è obbligatorio se è presente un evento finale;
     - **evento finale:** indica la conclusione di un processo, può essere opzionale, possono essercene più di uno ed è obbligatorio se è presente un evento iniziale.
   - **Activity:** è una attività da svolgere durante il corso di un business process. Una attività può essere atomica (*task*) o composta (*sub-process*). È rappresentata da un box arrotondato;
   - **Gateway:** è utilizzato per controllare la divisione e l'unione di percorsi nelle sequenze dei flussi. È rappresentato con un rombo. Questi si dividono in:
     - **decisionale:** in base alle condizioni il flusso continua la sua esecuzione in un solo ramo di uscita (if). Quando si riuniscono, si attende il completamento di un solo ramo in entrata prima di attivare il flusso in uscita;
     - **parallelo:** viene usato per separare il flusso di sequenza, tutti i rami uscenti vengono attivati contemporaneamente. Quando si riuniscono rami paralleli è necessario attendere il completamento di tutti i rami prima di attivare il flusso in uscita.

4. **Connecting objects**, che si dividono in:

   - **Sequence flow:** mostra l'ordine in cui le attività devono essere eseguite. È rappresentato da una linea continua con una punta di freccia piena;

- **Message flow:** mostra il flusso dei messaggi che inviano due diversi partecipanti di un processo. È rappresentato da una linea tratteggiata con la punta della freccia vuota;
- **Association:** viene utilizzato per associare dati, testo e altri manufatti con oggetti del flusso.

2.2 Log

Il termine **Log**, in ambito informatico, indica un file sequenziale che viene utilizzato per registrare, in modo cronologico, tutte le operazioni che vengono eseguite. Spesso questo tipo di file viene chiuso e conservato secondo delle scadenze programmate, ad esempio giornaliera, settimanale, mensile, ecc...

I log possono essere utilizzati per svariati scopi:

- analisi delle segnalazioni di errore;
- produzione di statistiche di esercizio, come ad esempio quelle del traffico nei servizi web;
- ripristino di situazioni precedenti, nel caso si fosse verificata una anomalia;
- analisi delle modifiche apportate alla base di dati;
- analisi delle operazioni eseguite e dei responsabili di tali operazioni;
- riassunto di quanto successo in un determinato arco di tempo.

Il log può memorizzare al suo interno anche una chiave cronologica (timestamp) e può rappresentare anche una porzione di una base di dati mantenendo comunque la sua funzionalità di registro cronologico.

All'interno di un sistema di elaborazione dati coesistono diverse tipologie di log:

- **Log di sistema:** questa tipologia è utilizzata frequentemente nei server di rete. In particolare vengono registrati gli eventi significativi riguardanti il flusso di dati tra il sistema, nel ruolo di fornitore di servizi e di applicazioni, ed i clienti che utilizzano tali servizi;
- **Log di applicazione:** molte applicazioni generano in maniera automatica dei log, indipendenti dal sistema. All'interno di essi vengono registrati gli eventi caratteristici dell'applicazione;

- **Log di base di dati:** questa tipologia riguarda la registrazione di tutte le operazioni che vengono effettuate su una base di dati (inserimento, modifica e cancellazione di record). Di questa registrazione se ne occupa il sistema gestore della base di dati stessa (DBMS). Nelle basi di dati più evolute, che forniscono servizi di tipo transazionale, il log viene utilizzato come riferimento per eseguire una *commit* (transazione completata) oppure una *rollback* (transazione annullata);

- **Log di sicurezza:** al suo interno vengono registrate tutte le operazioni che sono considerate critiche per l'integrità dei dati; viene tenuta traccia di tutti i tentativi di accesso al sistema, in modo da scoprire eventuali accessi non autorizzati. Questa tipologia trova un ampio utilizzo nei sistemi informatici complessi che contengono al loro interno dati particolarmente sensibili;

- **Log di eventi (event log):** al suo interno vengono registrate tutte le attività svolte all'interno di un processo aziendale. Approfondiremo questo tipo di log nella Sezione 2.3.4.

2.3 Process Mining

A causa dell'utilizzo sempre più massivo di sistemi informativi e dell'espansione del campo di applicazione di questi ultimi tra il parco di attività amministrative, economiche e produttive, si è andata amplificando anche la mole di dati prodotta per registrare ed analizzare tutte queste attività. Questi dati vengono memorizzati all'interno dei log.

L'idea di base del process mining è quella di modellare, monitorare e migliorare i processi estraendo conoscenza dai log. I log contengono infatti informazioni legate all'esecuzione dei processi nel mondo reale che sono di vitale importanza per la definizione di strategie per migliorare la qualità dei processi e ridurne i costi.

2.3.1 Che cos'è?

Il Process Mining costituisce un'area di ricerca relativamente giovane, che si situa, da un lato, tra la computational intelligence ed il data mining e, dall'altro, tra la modellazione e l'analisi dei processi. L'idea di base del Process Mining è quella di dedurre, monitorare e migliorare i processi reali (cioè non ipotetici) estraendo conoscenza dai log, che sono oramai ampiamente disponibili nei sistemi informativi.

Il Process Mining comprende:

- la deduzione (automatica) di processi, cioè l'estrazione di un modello di processo a partire da un log;
- la verifica di conformità, cioè il monitoraggio di eventuali discrepanze tra un modello ed un log;
- l'individuazione di reti sociali (social network) ed organizzative;
- la costruzione automatica di modelli di simulazione;
- l'estensione e la revisione di modelli;
- la predizione delle possibili future evoluzioni di un'istanza di processo;
- le raccomandazioni su come operare sulla base di dati storici.

Il Process Mining fornisce un importante punto di contatto fra data mining, modellazione ed analisi di processi.

La nuova sfida è cercare di sfruttare i dati in modo significativo, per esempio per fornire suggerimenti, identificare colli di bottiglia, prevedere problemi, registrare violazioni di regole, raccomandare contromisure e dare "forma" ai processi. Lo scopo del Process Mining è fare esattamente questo.

Il punto di partenza per qualsiasi tecnica di Process Mining è un log degli eventi (event log o semplicemente log). Tutte le tecniche di Process Mining assumono che sia possibile registrare sequenzialmente eventi in modo che ogni evento si riferisca ad una determinata attività (cioè ad un passo ben definito di un processo) e sia associato ad un particolare case (cioè un'istanza di processo). I log possono contenere anche ulteriori informazioni circa gli eventi. Di fatto, qualora sia possibile, le tecniche di Process Mining usano informazioni supplementari come le risorse (cioè le persone o i dispositivi) che eseguono o che danno inizio ad un'attività, i timestamp o altri dati associati ad un evento (per esempio la dimensione di un ordine).

Esistono alcuni fraintendimenti comuni quando si parla di Process Mining. Alcuni produttori, analisti e ricercatori tendono a pensare alle tecniche di Process Mining come a speciali approcci di Data Mining per il discovery di processi che si limitano ad un'analisi offline. Tuttavia, questa visione non è corretta. Pertanto, evidenziamo tre elementi caratterizzanti il Process Mining.

Per contestualizzare il Process Mining, consideriamo il ciclo di vita del Business Process Management (BPM). Il ciclo di vita BPM mostra le sette fasi di un processo di business ed i sistemi informativi sottostanti. Nella fase di

(ri)modellazione si crea un nuovo modello di processo, oppure se ne aggiorna uno preesistente. Nella fase di analisi, un processo candidato ed una possibile alternativa sono messi a confronto. Terminata la fase di (ri)modellazione, si implementa il modello (fase di implementazione) oppure, si (ri)configura un sistema informativo già esistente (fase di (ri)configurazione). Nella fase di esecuzione, il modello di processo è eseguito. Durante l'esecuzione, il processo è monitorato. Minori adattamenti sono possibili (fase di adattamento) senza che sia realizzata una nuova modellazione del processo. Vedi Figura 2.2 [van der Aalst and Adriansyah, 2012]. Nella fase di diagnosi, l'esecuzione del processo viene analizzata e questo può portare ad una nuova fase di rimodellazione. Il Process Mining è uno strumento utile per la maggior parte delle fasi descritte in precedenza. Ad esempio, durante l'esecuzione, le tecniche di Process Mining possono essere utilizzate come supporto alle decisioni (operational support). Si possono utilizzare per esempio strumenti di predizione e raccomandazione (che apprendono modelli sulla base di dati storici) per modificare le istanze di processo in esecuzione. Forme simili di supporto alle decisioni possono aiutare ad adattare il processo e a guidare la rimodellazione dello stesso.

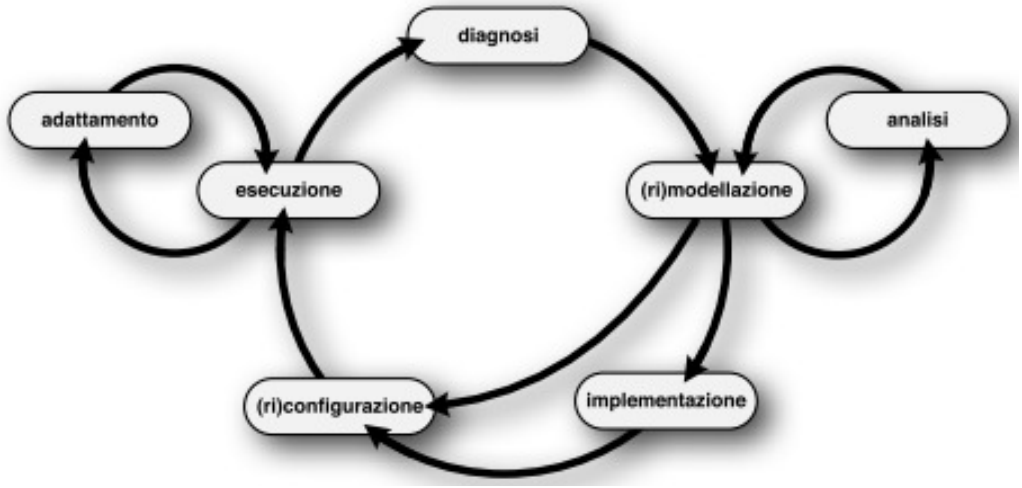

Figura 2.2: Il Ciclo di Vita BPM

La Figura 2.3 [van der Aalst and Adriansyah, 2012] analizza le attività ed i risultati del Process Mining, sono infatti riportate le possibili fasi in cui si può trovare un progetto di Process Mining. Ogni progetto inizia con una pianificazione ed una giustificazione dell'attività di pianificazione stessa (fase 0). Dopo l'avvio del progetto, è necessario interrogare sistemi informativi, esperti di dominio e manager per ricavare dati, modelli, obiettivi e domande a

cui è necessario successivamente rispondere (fase 1). Questa attività richiede una comprensione dei dati che si hanno a disposizione ("quali dati si possono usare per l'analisi?") e del dominio ("quali sono le domande rilevanti?") e fornisce i risultati (dati storici, modelli progettati, obiettivi, domande). Durante la fase 2, si costruisce il modello del flusso di controllo e lo si collega al log. In questa fase si possono usare tecniche di discovery automatico. Il modello dedotto può già fornire risposta ad alcune delle domande poste e potrebbe portare ad adattamenti e rimodellazioni. Inoltre, il log può essere filtrato o adattato sulla base del modello (ad esempio, rimuovendo attività rare o istanze anomale ed inserendo eventi mancanti). Talvolta è richiesto ulteriore lavoro per correlare gli eventi appartenenti alla stessa istanza di processo. Gli eventi rimanenti sono correlati ad entità del modello di processo. Quando il processo è piuttosto strutturato, il modello del flusso di controllo può essere esteso con altre prospettive (ad esempio, dati, tempi, risorse) durante la fase 3. La relazione stabilita durante la fase 2, tra log e modello, può essere usata per estendere il modello stesso (ad esempio, i timestamp di eventi associati possono essere utilizzati per stimare i tempi di attesa delle attività). Ciò può essere usato per rispondere ad ulteriori domande e può far scaturire nuove azioni. Infine, i modelli costruiti durante la fase 3 si possono utilizzare per il supporto alle decisioni (fase 4). La conoscenza estratta a partire dai dati storici sugli eventi è combinata con informazioni riguardanti le istanze in esecuzione. In questo modo è possibile modificare, predire e raccomandare. È opportuno evidenziare che le fasi 3 e 4 possono essere eseguite solo se il processo è sufficientemente stabile e strutturato. Attualmente, esistono tecniche e strumenti che permettono di realizzare tutte le fasi riportate in Figura 2.3 [van der Aalst and Adriansyah, 2012]. Tuttavia il Process Mining è un paradigma relativamente nuovo e la maggior parte degli strumenti disponibili non sono maturi. Inoltre, i nuovi utenti spesso non sono a conoscenza del potenziale e delle limitazioni del Process Mining [van der Aalst and Adriansyah, 2012].

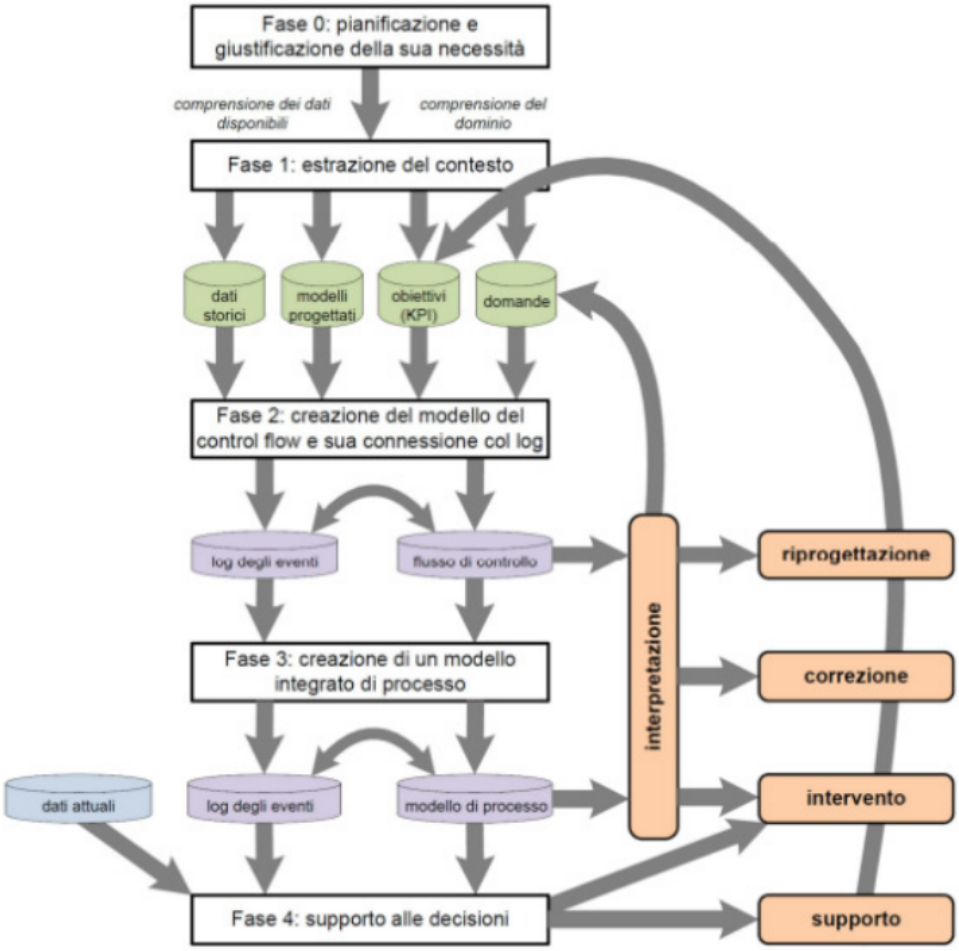

Figura 2.3: Ciclo di vita di un modello che descrive un progetto di Process Mining

2.3.2 Tecniche di Process Mining

Come mostrato nelle Figure 2.4 e 2.5, gli event log possono essere usati per eseguire tre tipi di Process Mining [van der Aalst and Adriansyah, 2012]:

1. **Discovery:** la tecnica di discovery prende in input un event log e produce un modello senza utilizzare alcuna informazione a priori. Il process discovery è la più importante tecnica di Process Mining, e molte organizzazioni che lo hanno utilizzato rimangono stupide dal fatto che queste tecniche riescano effettivamente a descrivere i processi reali solamente basandosi su esempi di esecuzione;

2. **Conformance Checking:** un modello di processo preesistente è confrontato con informazioni (relative allo stesso processo) estratte da un event log. Il conformance checking può essere usato per verificare se ciò che accade nella realtà (come risulta dai log) è conforme al modello e viceversa. Il conformance checking può essere applicato a diversi tipi di modelli: procedurali, organizzativi, dichiarativi, regole di business, leggi, etc;

3. **Enhancement:** l'idea è quella di estendere o migliorare un modello di processo esistente usando informazioni, relative al processo, contenute nei log. Mentre il conformance checking misura quanto un modello è allineato con ciò che accade nella realtà, questo terzo tipo di Process Mining si propone di cambiare o estendere il modello preesistente. Per esempio, usando i timestamp in un event log, è possibile estendere il modello per mostrare colli di bottiglia, livelli di un servizio, tempi di produttività e frequenze.

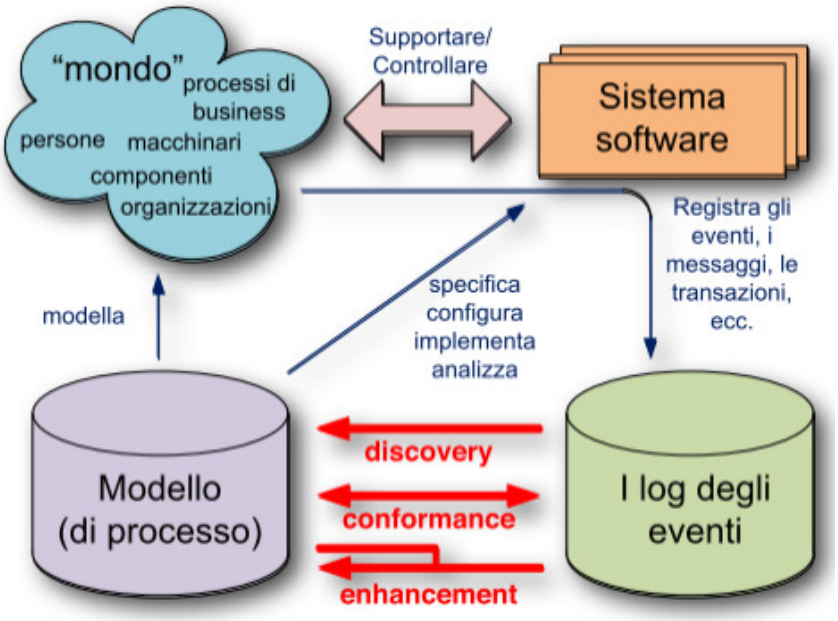

Figura 2.4: I principali tipi di Process Mining

La Figura 2.5 [van der Aalst and Adriansyah, 2012] mostra i tipi di input e di output delle 3 tecniche di Process Mining. Le tecniche di discovery prendono in input un event log e producono un modello. Il modello estratto è tipicamente un modello di processo (per esempio una rete di Petri, un modello BPMN o un diagramma UML delle attività). Tuttavia, il modello può anche descrivere altre prospettive (come per esempio un social network). Le tecniche di conformance checking prendono in input un event log ed un modello. L'output consiste in una serie di informazioni diagnostiche che mostrano le differenze tra il modello e il log. Anche le tecniche di enhancement (miglioramento), infine, richiedono un event log ed un modello in input. L'output è il modello stesso esteso con informazioni riguardanti ad esempio il tempo delle attività.

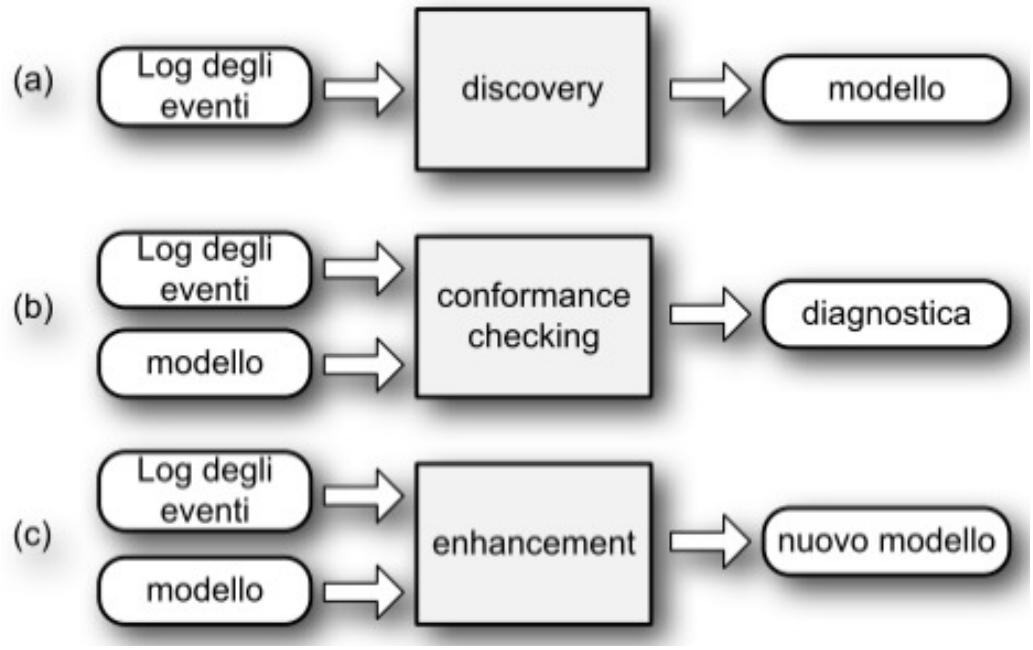

Figura 2.5: Input ed output delle 3 tecniche di Process Mining

2.3.3 Prospettive del Process Mining

Il Process Mining include varie prospettive [van der Aalst and Adriansyah, 2012]:

- **La prospettiva del flusso di controllo:** si focalizza sull'ordine delle attività. L'obiettivo di questa prospettiva consiste nel trovare una

buona caratterizzazione di tutti i possibili percorsi in un processo. Il risultato, tipicamente, è espresso in termini di una rete di Petri o di altri formalismi (come per esempio BPMN o diagrammi UML delle attività).

- **La prospettiva dell'organizzazione:** si focalizza sulle informazioni che riguardano le risorse contenute (e spesso non visibili) all'interno dei log, ovvero, quali attori (per esempio persone, sistemi, ruoli o dipartimenti) sono coinvolti e come questi si relazionano fra loro. L'obiettivo è quello di strutturare l'organizzazione classificando le persone in base ai ruoli che svolgono e alle unità organizzative, oppure di costruire una rappresentazione del social network dell'organizzazione.

- **La prospettiva dell'istanza:** si concentra sulle proprietà di una singola esecuzione (detta "case" o "traccia"). Ovviamente, un case può essere caratterizzato dal suo percorso nel processo oppure dagli attori che operano nello stesso. Tuttavia, i case possono essere definiti anche attraverso valori assunti da altri tipi di dati. Ad esempio, se un case rappresenta la compilazione di un ordine, potrebbe essere interessante conoscere il fornitore o il numero di prodotti ordinati.

- **La prospettiva del tempo:** è legata a quando un evento è accaduto e alla sua frequenza. Quando agli eventi sono associati dei timestamp, è possibile individuare colli di bottiglia, misurare i livelli di un servizio, monitorare l'uso delle risorse e predire il tempo restante per il completamento di un'istanza.

2.3.4 Event Log

Il Process Mining basa tutta la sua analisi sull'utilizzo di event log, particolari log che registrano al loro interno tutte le attività che vengono svolte in un determinato ambito aziendale. Spesso racchiudono informazioni relative alla risorsa che ha eseguito una determinata azione, al suo ruolo o al reparto cui appartiene, il timestamp di tale azione, ecc. Ogni azione solitamente ha un ciclo di vita, tipicamente composto da una fase di start ed una di complete (il ciclo di vita può essere anche più complesso), che indicano generalmente l'inizio e la conclusione di quella operazione.

Come possiamo vedere nella Tabella 2.1, un event log è composto da una serie di record che vengono registrati in maniera sequenziale. Nel nostro esempio il campo "Case ID" è la chiave di correlazione dell'event log, in quanto ci permette di mettere in relazione le istanze di un processo all'interno del

Case ID	Task Name	Event Type	Resource	Timestamp	Miscellaneous
1	Registrazione	Start	Gabriele	20-06-2014 14:00:00	...
1	Registrazione	Complete	Gabriele	20-06-2014 15:00:00	...
1	Aereo	Start	Gabriele	20-06-2014 15:05:00	...
2	Registrazione	Start	Giorgio	20-06-2014 15:07:00	...
2	Registrazione	Complete	Giorgio	20-06-2014 15:50:00	...
1	Aereo	Complete	Gabriele	20-06-2014 17:00:00	...
1	Prenotazione	Start	Gabriele	22-06-2014 11:00:00	...
1	Prenotazione	Complete	Gabriele	22-06-2014 11:10:00	...
2	Treno	Start	Giorgio	24-06-2014 15:05:00	...
2	Treno	Complete	Giorgio	24-06-2014 15:06:00	...

Tabella 2.1: Esempio di Event Log

log. Gli altri campi indicano invece il nome dell'attività, il tipo di evento che si è verificato (lifecycle attività), la risorsa che ha svolto tale attività, la data e l'ora di esecuzione. La chiave di correlazione risulta molto importante, vista la struttura sequenziale del log, in quanto è l'unico modo che ci permette di analizzare l'evolversi di una istanza all'interno del log stesso.

Nel 2010, è stato anche standardizzato XES, un formato per la memorizzazione dei log, estensibile e supportato dalla libreria OpenXES, che è presentato nella Sezione 2.5.

2.4 ProM

Parte del successo del Process Mining deriva dallo sviluppo di ProM, che è un framework Open Source per il supporto di una grande quantità di tecniche di Process Mining. ProM [Verbeek and Günther, 2014] è un progetto del Process Mining Group della Eindhoven Technical University [2]. Fornisce una piattaforma per utenti e sviluppatori di algoritmi di Process Mining facile da usare e da estendere. Implementato in Java, è modulare e distribuito con una licenza GNU Public License (GPL). ProM è arrivato alla versione 6.3, parallelamente è disponibile la versione Nightly, che permette di avere sempre l'ultima versione del framework disponibile. ProM è modulare perché gli algoritmi di Process mining sono sviluppati sotto forma di plug-in di ProM, con notevoli vantaggi per chi deve implementare una nuova tecnica. ProM si compone del framework vero e proprio e di un Package Manager, un tool attraverso il quale è possibile installare nuovi plug-in e verificare tutti gli aggiornamenti di quelli già installati.

[2] http://www.promtools.org/prom6/

```
# Specifies the ProM release version
PROM_VERSION = 6.3.0
#
# Specifies which package should be installed
RELEASE_PACKAGE = EstablishedPackages
#
# Specifies the URL to the default package repository
# (default is "http://www.promtools.org/prom6/packages63/packages.xml")
PACKAGE_URL = http://www.promtools.org/prom6/packages/packages.xml
```

Figura 2.6: File di configurazione di ProM

Attraverso il file di configurazione ProM.ini è possibile specificare moltissimi parametri di configurazione dell'applicazione ed in particolare possiamo specificare l'indirizzo del package repository. Attraverso questa impostazione possiamo decidere da dove andare a scaricare i nostri pacchetti per i plug-in. Esiste infatti la possibilità di impostare il repository classico, contenente le versioni testate e rilasciate oppure scaricare le versioni contenenti gli ultimi aggiornamenti rilasciati in fase di sviluppo. Come mostra la Figura 2.6, noi abbiamo deciso di utilizzare le versioni più aggiornate, in fase di sviluppo, per la nostra analisi.

Figura 2.7: Scheda Workspace ProM

Attualmente, ci sono già più di 120 packages contenenti più di 500 plug-in. Questi plug-in possono essere divisi in varie tipologie. Esistono dei plug-in per l'import/export di modelli di processo, come ad esempio:

- Petri nets (PNML, TPN);

- EPCs / EPKs (EPML);
- BPMN (XPDL, BPMN2);
- Process Trees (PTML).

Ci sono plug-in di mining, come ad esempio:

- Plug-in che supportano le tecniche per il mining del modello del processo (come l'algoritmo Alpha e il Genetic Mining);
- Plug-in che analizzano i log dal punto di vista organizzativo (come il Social Network miner);
- Plug-in per il mining di processi flessibili (come il Fuzzy Miner);
- Plug-in per il mining di processi strutturati ad albero (come il Evolutionary Tree miner ed Inductive Miner).

Inoltre, ci sono i plug-in di analisi che si occupano di:

- Verifica dei modelli di processo (ad esempio, Replay a Log on Petri Net for Conformance Analysis);
- Verifica di proprietà specificate in Linear Temporal Logic (LTL) formulate rispetto log;
- Analisi delle prestazioni (Basic statistical analysis e Performance Analysis);
- Supporto per la configurazione di modelli di processo configurabili.

Infine, ProM sfoggia una serie di filtri di log, che sono uno strumento prezioso per la pulizia di quest'ultimi da artefatti indesiderati, o poco importanti.

La Figura 2.7 mostra l'interfaccia utente di Prom. Come possiamo osservare sono presenti 3 schede, contrassegnate rispettivamente da numeri 1, 2 e 3.

1. **scheda Workspace:** porta alla visualizzazione del lavoro. Questa visualizzazione mostra tutte le risorse, sia quelle della form principale che quelle frutto di elaborazioni. La Figura 2.7 mostra le parti principali di questa sezione:

   (A) il pulsante "Import" importa una risorsa da un file in ProM;

(B) un'anteprima della risorsa selezionata, se disponibile, con dettagli aggiuntivi, come il nome e il tipo;

(C) il pulsante "View", si raggiunge la scheda view per la risorsa selezionata;

(D) il pulsante "Azione" porta alla visualizzazione dell'azione con la risorsa selezionata aggiunta come input per l'azione;

(E) il pulsante "Rimuovi" rimuove la risorsa selezionata;

(F) la risorsa non selezionata;

(G) la risorsa selezionata. Il riquadro di visualizzazione delle risorse (I) mostra i dettagli su questa risorsa;

(H) il pulsante "Export to disk" esporta una risorsa da ProM in un file;

(I) il riquadro di vista delle risorse. Mostra dettagli sulla risorsa selezionata (se presente).

2. **scheda Actions:** porta alla vista della Action. Come mostrato in Figura 2.8, questa visualizzazione mostra tutte le azioni che le risorse possono prendere in input ed in output. La Figura 2.8 mostra le parti principali di questa sezione:

Figura 2.8: Scheda Actions ProM

(A) indica che la risorsa corrispondente è nel pool di risorse di input corrente;

(B) il pulsante "Workspace" porta alla vista workspace della corrispondente risorsa selezionata;

(C) il pulsante "Rimuovi" rimuove la risorsa corrispondente dal pool di risorse di input. Si noti che questo non rimuoverà la risorsa stessa, rimuove solo la risorsa da questo pool;

(D) la risorsa segnaposto consente di aggiungere una risorsa al pool di risorse di input. Questo segnaposto è disponibile solo se nessuna azione è stata selezionata nel pool delle action (F);

(E) il riquadro di input mostra il pool di risorse di input. Se viene avviata una action, allora queste risorse potranno essere utilizzate come input per la action;

(F) il riquadro action mostra il pool della action, che elenca le possibili action basate sia sul pool di risorse di ingresso (E) che sul pool dei tipi di output (I), e sul filtro delle action (G);

(G) il campo "Cerca" consente di filtrare il nome delle action;

(H) il tipo di segnaposto consente di aggiungere un tipo al pool dei tipi di output. Questo segnaposto è disponibile solo se nessuna azione è stata selezionata nel pool delle action (F);

(I) il riquadro output mostra il pool dei tipi di output. Se una action viene completata, allora verranno generate risorse di questi tipi;

(L) ll pulsante "Reset" ripristina la vista della action: azzera il pool di risorse di input, il pool dei tipi di output, il filtro e deseleziona tutte le action nel pool di action;

(M) il pulsante "Start" avvia la action selezionata sulle risorse di input selezionate.

3. **scheda View:** porta alla form "Vista". Come mostrato in Figura 2.9, questa sezione mostra una risorsa o una panoramica di tutte le risorse per le quali esiste una visione:

(A) l'elenco a discesa consente di selezionare visioni alternative (sulla stessa risorsa);

(B) il pulsante "Aggiorna" aggiorna la vista corrente;

(C) il pulsante "Stampa" consente di stampare la vista corrente;

Figura 2.9: Schermata di visualizzazione grafica di ProM

(D) il pulsante "Action" porta alla visualizzazione action con la risorsa corrispondente aggiunta come input per la action;

(E) il pulsante "Workspace" porta alla vista workspace con la corrispondente risorsa selezionata;

(F) il pulsante "Anteprima" porta alla vista panoramica, mostrata in Figura 2.10;

(G) in questa area vengono elencate le statistiche relative alla risorsa che stiamo analizzando;

(H) l'area principale mostra la vista sulla risorsa corrispondente.

Il sito di riferimento di ProM è http://www.promtools.org/prom6/, dal quale è possibile scaricare le due versioni sopra descritte [Verbeek, 2010a] [Verbeek, 2010b].

2.5 Il formato XES

Nel 2011 la Task Force IEEE del Process Mining ha deciso di adottare il formato XES (eXtensible Event Stream) come standard per l'archiviazione dei log di eventi. Xes è anche diventato il formato standard nella versione 6 del framework ProM.

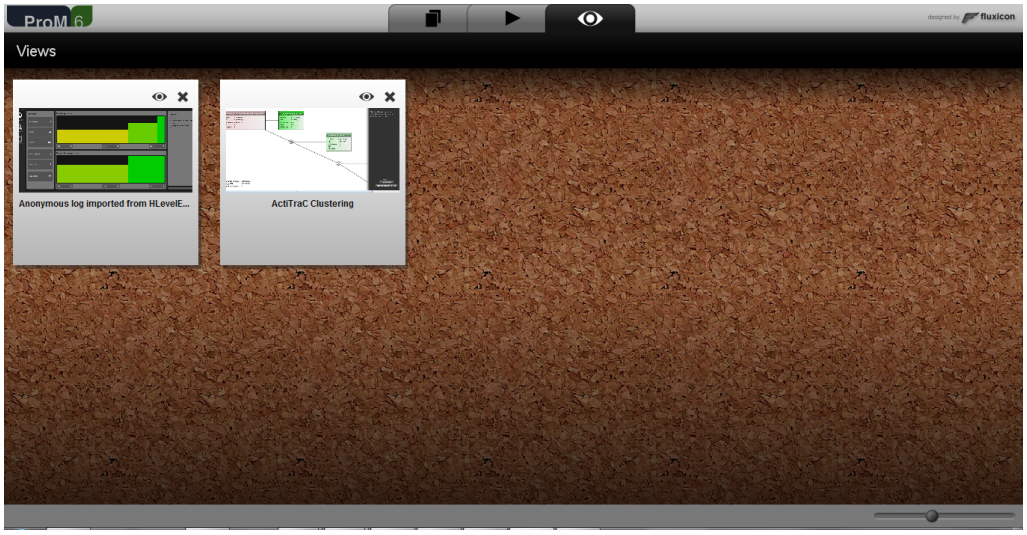

Figura 2.10: Schermata di riepilogo sezioni aperte di ProM

Questo formato è stato sviluppato per superare i limiti del precedente linguaggio utilizzato nei tool di Process Mining, il formato MXML (Macromedia Flex Markup Language, è un linguaggio dichiarativo basato su XML).

XES [Günther and Verbeek, 2014] è una standard basato sull'XML (eXtensible Markup Language). Il suo scopo è quello di fornire un formato generalmente riconosciuto per l'interscambio di dati dell'event log tra strumenti e dominio di applicazione. Il suo scopo principale è per il process mining , vale a dire l'analisi dei processi operativi basati sui loro event log. XES è stato progettato per essere adatto anche per il generico data mining, text mining ed analisi statistica.

Nel progettare lo standard XES, sono stati utilizzati, come principi guida, i seguenti obiettivi:

- **Semplicità:** utilizzare il modo più semplice possibile per rappresentare le informazioni. I log XES dovrebbero essere facili da analizzare e generare, e dovrebbero essere ugualmente ben leggibili;
- **Flessibilità:** lo standard XES dovrebbe essere in grado di catturare gli event log da qualsiasi contesto, indipendentemente da quale sia il dominio di applicazione o il supporto informatico del processo osservato. Così, XES mira a guardare oltre il process mining ed i business process, e si sforza di essere uno standard per i dati dell'event log;
- **Estensibilità:** lo standard deve essere facilmente estendibile. L'estensione dello standard dovrebbe essere il più trasparente possibile,

pur mantenendo la compatibilità all'indietro e in avanti. Allo stesso modo, deve essere possibile estendere lo standard per esigenze particolari, ad esempio per i domini applicativi specifici, o per specifiche implementazioni di strumenti;

- **Espressività:** nonostante la ricerca di un formato generico, gli event logs serializzati in XES dovrebbero incontrare la minor perdita di informazioni possibile. Tutti gli elementi informativi devono essere fortemente tipizzati, e devono permettere di assegnare un significato non ambiguo a ciascun elemento.

Solo gli elementi che possono essere identificati in qualsiasi ambiente sono esplicitamente definiti dallo standard. Tutte le ulteriori informazioni sono rinviate ad attributi opzionali, che possono essere standardizzati (in termini di semantica) dalle estensioni esterne.

2.5.1 Il meta modello XES

Come mostrato in Figura 2.11 [Günther and Verbeek, 2014], la gerarchia di base di un documento XES segue la struttura universale delle informazioni del registro degli eventi:

- **Log:** al livello superiore è presente un oggetto di log, che contiene tutte le informazioni sugli eventi che sono legati ad un processo specifico. Il nome del tag per l'oggetto log nella serializzazione XML di XES è: `<log>`. Attributi del tag `<log>` XML sono:
  - xes.version
  - xes.features
- **Trace:** un *registro* contiene un numero arbitrario (può essere vuoto) di oggetti di traccia. Ogni traccia descrive l'esecuzione di un caso specifico, o il caso, del processo registrato. Il nome del tag per l'oggetto trace nella serializzazione XML di XES è: `<trace>`. Non sono definiti attributi XML per il tag `<trace>`.
- **Event:** ogni traccia contiene un numero arbitrario (può essere vuoto) di oggetti evento. Gli eventi rappresentano la granularità atomica delle attività che sono state osservate durante l'esecuzione di un processo. Come tale, un evento ha una durata. Il nome del tag per l'oggetto event nella serializzazione XML di XES è: `<event>` Non sono definiti attributi XML per il tag `<event>`.

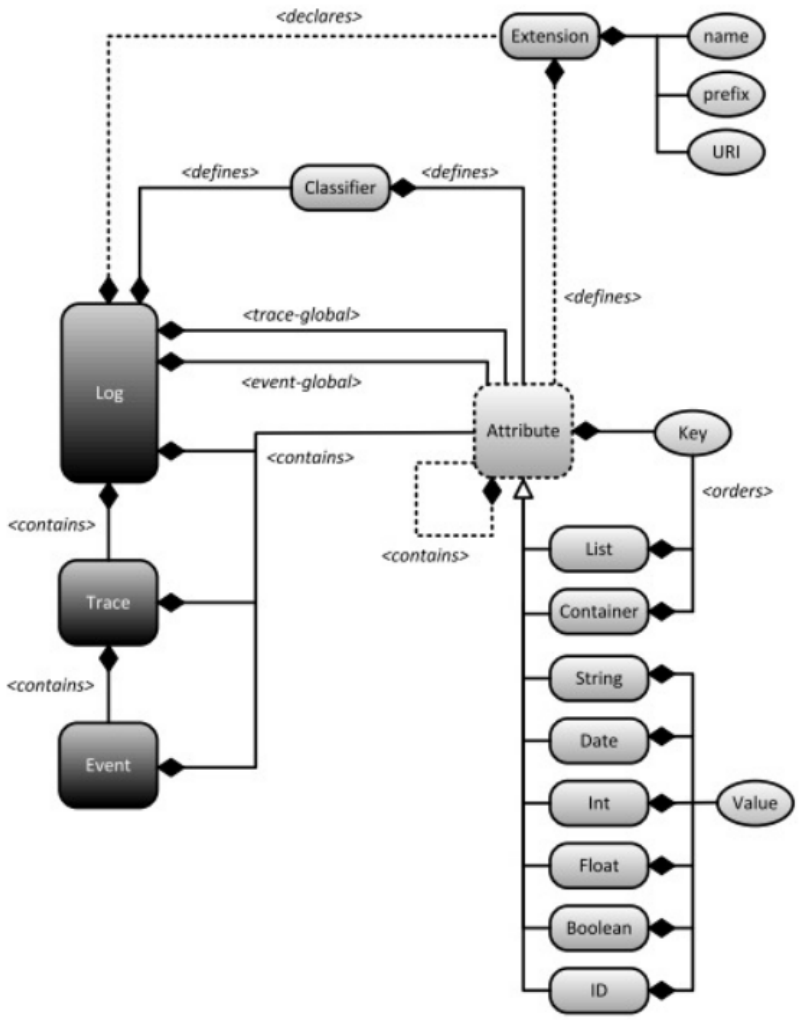

Figura 2.11: Il metamodello XES

Le tre tipologie di oggetti di log, appena elencate, non contengono informazioni. Esse definiscono solamente la struttura del documento. Tutte le informazioni, in un registro eventi, vengono memorizzate negli attributi. Gli attributi descrivono il loro elemento genitore (log, trace, ecc.) Tutti gli attributi hanno una chiave basata su stringa. Lo standard XES richiede che l'attributo chiave:

- non contenga avanzamenti riga, ritorni a capo e tabs. Può contenere spazi iniziali o finali, o più spazi;
- sia univoco nel contenitore che lo racchiude (ad esempio, un solo attributo con la chiave "id" per trace). L'unica eccezione a questa regola

sono le chiavi all'interno di un elenco di inclusione. Poiché l'elenco impone un ordine su queste chiavi, hanno bisogno di non essere uniche.

Logs, trace ed event contengono ciascuno un numero arbitrario di attributi. Ci sono sei tipi di attributi elementari, ciascuno definito dal tipo di valore di dati che rappresenta:

- **String:** gli attributi *String* contengono informazioni letterali di lunghezza arbitraria che non sono generalmente tipizzate. Nella rappresentazione XML di XES, i valori degli attributi *String* vengono memorizzati come `xs:string` come tipologia di dati.

  Un esempio:

  ```
  <string key="name" value="nome"/>
  ```

- **Date:** gli attributi *Date* contengono informazioni su uno specifico punto nel tempo (con precisione millisecondi). Nella rappresentazione XML di XES, i valori degli attributi *Date* vengono memorizzati come `xs:dateTime` come tipologia di dati.

  Un esempio:

  ```
  <date key="data" value="2014-05-25T21:35:47.342+01:00"/>
  ```

- **Int:** gli attributi *Int* contengono un numero intero discreto (con 64bit). Nella rappresentazione XML di XES, i valori degli attributi *Int* vengono memorizzati come `xs:long` come tipologia di dati.

  Un esempio:

  ```
  <int key="counter" value"162534"/>
  ```

- **Float:** gli attributi *Float* contengono un numero continuo in virgola mobile continuo (con 64 bit). Nella rappresentazione XML di XES, i valori dell'attributo *Float* vengono memorizzati come `xs:double` come tipologia di dati.

  Un esempio:

  ```
  <float key="percentage" value="17.43"/>
  ```

- **Boolean:** gli attributi *Boolean* contengono un valore booleano che può essere vero o falso. Nella rappresentazione XML di XES, i valori degli attributi *Boolean* vengono memorizzati come `xs:boolean` come tipologia di dati.

  Un esempio:

```
<boolean key="success" value="true"/>
```

- **ID:** gli attributi *ID* contengono informazioni sull'id che è generalmente un *universally unique identifier* (UUID). Nella rappresentazione XML di XES, i valori degli attributi *ID* vengono memorizzati come `xs:string` come tipologia di dati.

  Un esempio:

  ```
  <id key="customer" value="f81d4fae-7dec-11d0-a765-00a0c91e6bf"/>
  ```

Accanto a questi attributi elementari, ci sono due tipi di attributi di raccolta:

- **List:** gli attributi *List* contengono un numero qualsiasi (può essere vuoto) di attributi figli. Questi attributi figlio sono ordinati e le loro chiavi non hanno bisogno di essere univoche. Il valore di un attributo *List* è derivato dai valori dei suoi attributi figli.

  Un esempio:

  ```
  <list key="revision">
  <string key="name" value="XES standard"/>
  <boolean key="stable" value="true"/>
  <string key="revision" value="2.0"/>
  <string key="revision" value="1.4"/>
  </list>
  ```

- **Container:** gli attributi *Container* contengono un numero qualsiasi (può essere vuoto) di attributi figli. Gli attributi figli non sono ordinati. Il valore di un attributo *Container* deriva dai valori dei suoi attributi figli.

  Un esempio:

  ```
  <container key="location">
  <string key=""street" value="Provinciale Pisana"/>
  <int key="number"" value="541"/>
  <string key="cap" value="57121"/>
  <string key="city" value="Livorno"/>
  <string key="country" value="Italy"/>
  </container>
  ```

Oltre a questi tipi di attributi standard esistono le estensioni, che consentono di aggiungere ulteriori informazioni in un event log. Alcune estensioni sono state standardizzate, in quanto, certe informazioni, risultavano necessarie per quasi ogni event log. Queste estensioni standardizzate sono le seguenti e ciascuna si riferisce ad un tipo diverso di dato:

- **Concept:** è l'identificativo delle attività la cui esecuzione ha generato l'evento;
- **Lifecycle:** serve ad indicare se è un evento è solo iniziato (start) oppure si è già concluso (complete);
- **Organizational:** è utile quando gli eventi sono stati realizzati da una persona che in qualche modo fa parte di una struttura organizzativa. Questa estensione comprende la risorsa, il ruolo e il gruppo;
- **Time:** definisce la data e l'ora esatta in cui sono stati registrati gli eventi. Il timestamp nella registrazione degli eventi è fondamentale per la maggior parte delle analisi;
- **Semantic:** è il riferimento ai concetti del modello in una ontologia.

2.6 Riassunto

In questo capitolo sono stati descritti tutti gli argomenti trattati all'interno del progetto di tesi, cercando di focalizzare maggiormente l'attenzione sul Process Mining e sulle sue 3 tecniche principali. Sono stati inoltre spiegati i processi di business, compresa la notazione BPMN, ed il tool utilizzato per le eseguire le varie analisi necessarie per la realizzazione di questo progetto di tesi.

Capitolo 3

CONTESTO DEI DATI

In questo capitolo verrà introdotto il contesto dei dati che sono stati utilizzati per le tecniche di process mining presentate in questo lavoro di tesi. In particolare vengono descritte le esigenze che hanno portato allo sviluppo di una piattaforma che raccogliesse i dati relativi all'importazione e all'esportazione delle merci.

3.1 TPCS

In Italia viene prodotta una dichiarazione doganale ogni due secondi, in totale 10,5 milioni di dichiarazioni all'anno. In media vengono emessi 68 documenti per ogni dichiarazione doganale e gli enti che si occupano dell'emissione di quest'ultimi sono circa 18. Analizzando soltanto la documentazione cartacea prodotta a Livorno, nel corso del tempo, questa ha raggiunto una mole di oltre 10 tonnellate.
Vista l'elevata quantità di informazioni, la digitalizzazione dell'intero processo di sdoganamento è diventato l'obiettivo primario da raggiungere per ridurre i costi e snellire le procedure burocratiche all'interno del porto di Livorno.

L'Autorità Portuale di Livorno ha deciso, attraverso la sottoscrizione di un Protocollo d'Intesa con l'Agenzia delle Dogane, lo sviluppo di un'iniziativa finalizzata a ridurre i tempi e velocizzare le procedure necessarie allo sdoganamento della merce, a beneficio della Comunità Portuale e di tutti gli operatori abilitati. La collaborazione tra i due enti ha portato alla nascita della piattaforma TPCS (Tuscan Port Community System).

TPCS è una piattaforma telematica basata sull'architettura web service e risulta essere uno strumento operativo efficace ed efficiente per le compa-

gnie di navigazione, per importatori ed esportatori, in quanto garantisce il controllo del percorso procedurale e fisico della merce, dal momento in cui parte sino al momento in cui arriva o viene imbarcata.

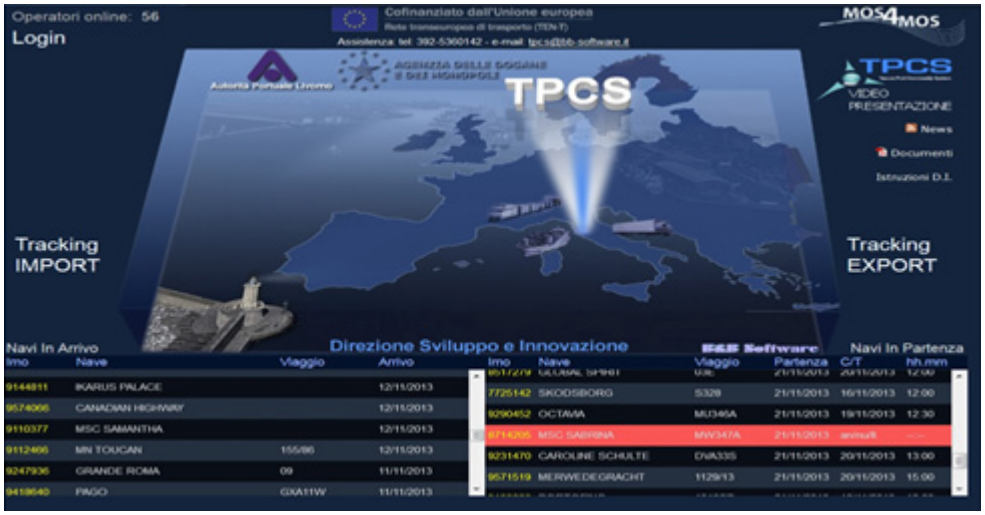

Figura 3.1: La piattaforma TPCS

TPCS permette:

- ai **vettori marittimi** di prelevare i dati relativi alla merce in partenza, in modo da poter presentare correttamente il Manifesto Merci in Partenza o Arrivo (MMP, MMA) in maniera completamente automatica, intervenendo solamente su eventuali errori o messaggi di diniego di carico;
- agli **spedizionieri merci** di presentare interattivamente la dichiarazione doganale e, a svincolo doganale ottenuto, le richieste d'imbarco elettroniche;
- ai **terminal portuali** di programmare ed effettuare in sicurezza gli imbarchi autorizzati, potendo importare da TPCS sui propri sistemi gestionali i dati di propria competenza;
- agli **uffici istituzionali** di monitorare, controllare e vigilare sull'intera attività portuale, in tempo reale, sia in entrata che in uscita;
- ai **trasportatori** di controllare dal TPCS lo stato delle partite di merce da ritirare presso i vari terminal e di stampare la distinta di uscita da esibire al gate.

TPCS garantisce la sicurezza dei dati mediante vari sistemi di archiviazione e recovery. È protetta a tre diversi livelli da accessi indesiderati e rispetta le norme della privacy commerciale. Ogni soggetto può visualizzare ed operare solamente sui dati di sua proprietà o strettamente necessari allo svolgimento delle sue attività.

Il sito di riferimento per questa piattaforma è http://www.tpcs.eu/.

3.1.1 Export

All'interno dell'Istituto di Scienza e Tecnologie dell'Informazione (ISTI) del CNR di Pisa, in collaborazione con l'Autorità Portuale, sono stati creati i modelli di Business per la descrizione dei processi di Import ed Export della piattaforma TPCS.

Per lo scopo di questa tesi viene utilizzato un modello semplificato del processo di Export della piattaforma TPCS, che è rappresentato in Figura 3.2.

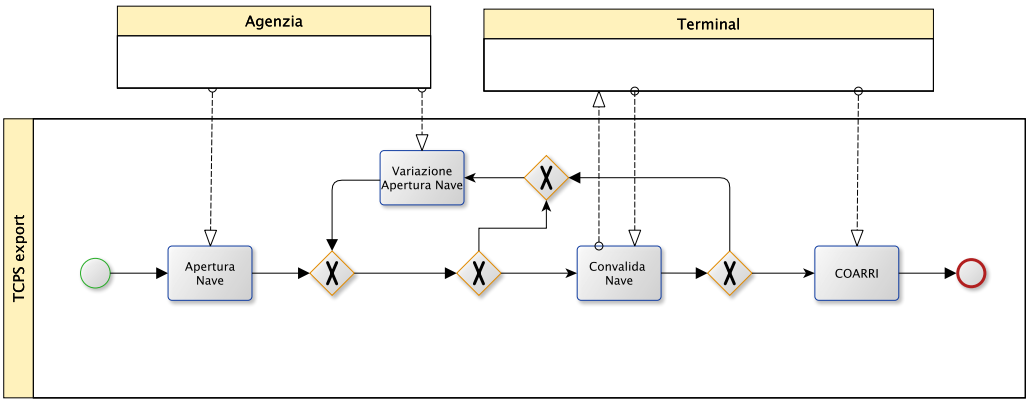

Figura 3.2: Modello del processo di Export della Tuscan Port Community System

TPCS gestisce i viaggi nave delle merci in esportazione o comunque destinate ad altri porti, monitorando le connesse applicazioni per la gestione dei Manifesti Merci in Partenza (MMP).

Come mostrato in Figura 3.2, il processo di Export è composto da 4 fasi principali:

1. **APERTURA NAVE:** l'agenzia provvede alla pratica di Apertura Nave, registrandola sul TPCS, dopo aver concordato con il terminal i parametri necessari (nome nave, viaggio, tempo stimato di arrivo e tempo stimato di termine operazioni);

2. **CONVALIDA NAVE:** il terminal effettua i controlli e poi provvede, in caso positivo, alla pratica di Convalida Nave, registrandola su TPCS;

3. **APERTURA NAVE-VARIAZIONE:** nel caso in cui la pratica di Apertura Nave, non sia stata effettuata correttamente, deve essere eseguita la pratica di Apertura Nave Variazione, che consiste nella modifica dell'Apertura Nave fatta inizialmente. Tale pratica può verificarsi sia prima che dopo la Convalida Nave, che comunque va effettuata;

4. **COARRI (Container discharge/loading report):** è un documento contenente la lista dei containers imbarcati/sbarcati da una nave. Questa fase è la parte conclusiva del processo di Export. Questa pratica viene effettuata dopo che sono state inviate la distinte di imbarco alla piattaforma TPCS.

3.2 Riassunto

In questo capitolo è stato descritto il contesto del processo che stiamo analizzato e cioè la piattaforma TPCS. In particolare sono state specificate le esigenze che hanno portato l'Autorità Portuale di Livorno a sviluppare questa piattaforma, nella quale vengono memorizzate tutte le operazioni che vengono effettuate dagli attori che partecipano attivamente al processo. Inoltre è stato mostrato e descritto il processo di Export, che è quello oggetto di tutta l'analisi.

Capitolo 4

PREPARAZIONE DATI

In questo capitolo verrà descritta la struttura base del file CSV contenente le registrazioni degli eventi. Verranno inoltre spiegate le modalità e gli strumenti utilizzati per eseguire la preparazione dei dati, soffermandosi in particolare sul plug-in usato per effettuare tale preparazione e sulle modifiche apportate a tale plug-in.

4.1 La struttura del log iniziale

La prima fase operativa del progetto consisteva nella preparazione dei dati in modo da poter essere adattati alle esigenze di analisi. I log che ci sono stati forniti erano in formato CSV (comma-separated values). CVS è un formato di file basato su testo che utilizza un separatore per distinguere i dati.

Un esempio di file CSV che utilizza il carattere "," come separatore, è dato dalla seguente tabella (dove i dati delle risorse, dei gruppi, ecc... sono frutto di fantasia per non rivelare dati sensibili):

id	Timestamp	Attività	Lifecycle	Resource	Group	Role	Nome Nave	IMO	Viaggio
511289	23/05/2014 08:41:58:827	APERTURA NAVE	START	SPANNA	MARE BLU	5t2n	LUPI DI MARE	9253143	15 A
511290	23/05/2014 08:41:58:858	APERTURA NAVE	COMPLETE	SPANNA	MARE BLU	5t2n	LUPI DI MARE	9253143	15 A
511291	23/05/2014 08:43:29:260	CONVALIDA NAVE	START	CIOCCI			LUPI DI MARE	9253143	15 A
511292	23/05/2014 08:43:29:338	CONVALIDA NAVE	COMPLETE	CIOCCI			LUPI DI MARE	9253143	15 A
511293	23/05/2014 10:03:33:850	APERTURA NAVE	START	LIPIDI	A.M. LIPIDI	MANN	SEA SHIP	9244881	MU421A
511294	23/05/2014 10:03:33:897	APERTURA NAVE	COMPLETE	LIPIDI	A.M. LIPIDI	MANN	SEA SHIP	9244881	MU421A
511295	23/05/2014 10:05:29:399	CONVALIDA NAVE	START	CIOCCI			SEA SHIP	9244881	MU421A
511296	23/05/2014 10:05:29:509	CONVALIDA NAVE	COMPLETE	CIOCCI			SEA SHIP	9244881	MU421A
511297	23/05/2014 10:05:37:512	APERTURA NAVE	START	LIPIDI	A.M. LIPIDI	MANN	MARE TRAVEL	8918978	MW422A
511298	23/05/2014 10:05:37:558	APERTURA NAVE	COMPLETE	LIPIDI	A.M. LIPIDI	MANN	MARE TRAVEL	8918978	MW422A

Tabella 4.1: Estratto del CSV utilizzato per l'analisi

La Tabella 4.1 mostra la struttura del file CSV che abbiamo utilizzato, i cui campi risultano essere i seguenti:

- **id:** è l'identificatore univoco della traccia sequenziale che è presente all'interno del log;
- **timstamp:** indica la data e l'ora di registrazione dell'evento;
- **attività:** indica il tipo di attività che è stata svolta;
- **lifecycle:** indica la fase dell'attività che è stata svolta;
- **resource:** indica quale risorsa ha effettuato l'operazione;
- **group:** indica il gruppo di appartenenza della risorsa che ha effettuato l'operazione;
- **role:** indica il ruolo della risorsa che ha effettuato l'operazione;
- **nomenave:** indica il nome della nave associata a quella spedizione;
- **imo:** indica il numero IMO (International Maritime Organization) e consiste in una sequenza di sette numeri assegnata ad ogni nave al momento della costruzione;
- **viaggio:** rappresenta un codice alfanumerico univoco che identifica il percorso della nave dal porto di partenza al porto di arrivo.

4.2 La trasformazione del log

Il formato di input ideale per le analisi di Process Mining è il formato XES, quindi abbiamo dovuto trasformare i dati da CSV a XES. Per effettuare questa trasformazione abbiamo esteso un plug-in presente all'interno di ProM, il *Convert Key/Value Set to Log*. Questo plug-in, che è stato sviluppato da Michael Westergaard, accetta in input un file CSV ed è in grado di trasformarlo nello standard XES. Per meglio adattarlo alle nostre esigenze, abbiamo modificato una parte del plug-in apportando delle migliorie. Nello specifico abbiamo inserito:

- la possibilità di includere dei campi aggiuntivi, presenti nel file CSV iniziale, al processo di conversione; basta infatti effettuare un doppio clic con il mouse sulla colonna che vogliamo inserire nel processo di conversione, selezionare la tipologia del dato ed in automatico il campo viene aggiunto con associata la corretta variabile;

- la possibilità di escludere un determinato campo dal processo di conversione, cliccando sull'icona del cestino a fianco della riga interessata;
- possibilità di scegliere il formato delle date, importantissimo per la gestione del campo *timestamp*;
- un servizio di messaggistica che aiuta l'utente a capire se le operazioni che ha eseguito sono andate a buon fine;
- un controllo che non permette l'inserimento di un campo già presente nella selezione dei campi interessati alla conversione.

Figura 4.1: Plug-in Convert Key/Value Set to Log

Di seguito vediamo come viene utilizzato il plug-in. Dopo aver avviato ProM ed aver importato il file CSV, basta selezionare il plug-in Convert Key/Value Set to Log, come è mostrato in Figura 4.1

Dopo aver cliccato sul pulsante "Start", si apre la schermata mostrata in Figura 4.2, che permette di effettuare il mapping dei campi. Come possiamo vedere, questa schermata è suddivisa in due sezioni principali:

1. la parte superiore, contenente una piccola anteprima del file CSV caricato;

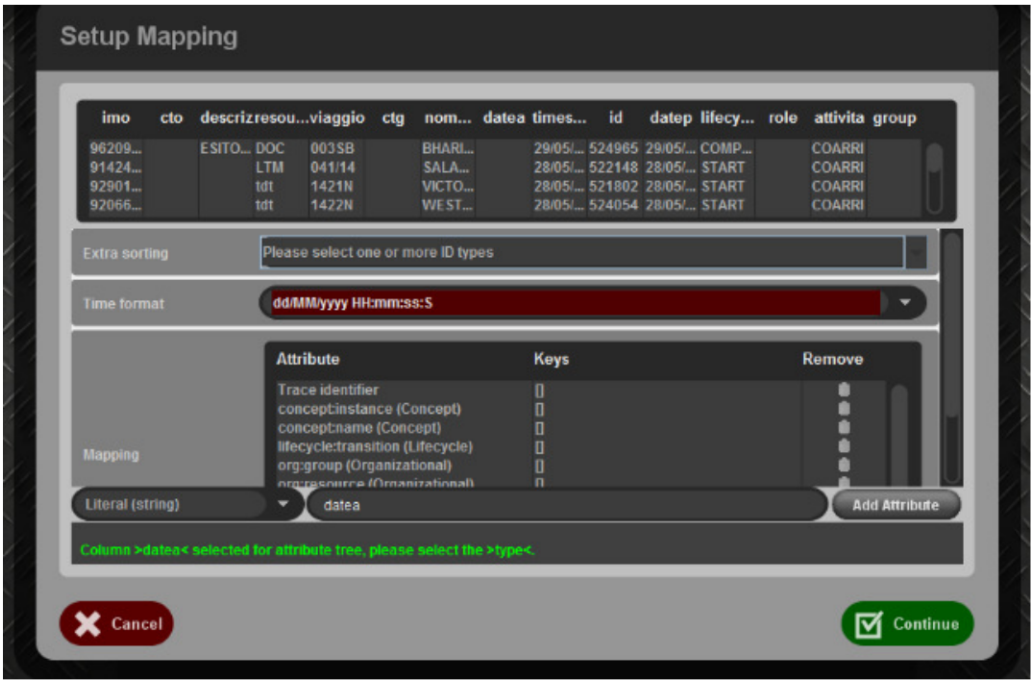

Figura 4.2: Sezione mapping del Plug-in Convert Key/Value Set to Log

2. la parte inferiore, contenente il mapping vero e proprio. Questa sezione permette di assegnare a ciascun attributo di un registro di eventi XES, il corrispondente attributo nel file di input e di aggiungere attributi personalizzati. È sufficiente selezionare l'identificatore di ciascuna traccia e l'identificatore degli eventi di ciascuna traccia con tutti i suoi attributi.

Una volta selezionati tutti i campi e controllato le corrette associazioni, cliccando sul pulsante "Continue" si apre una schermata contenente una panoramica del log e sarà possibile esportare il file XES generato, in modo da poter procedere con la fase di analisi vera e propria dei log. La decisione di inserire, nel processo di conversione, tutti i campi presenti nel log iniziale è stata presa per ottenere un file XES contenente il maggior numero di informazioni possibili in modo da poter fare delle analisi più approfondite.

Il plug-in cosi esteso è disponibile nel repository git all'indirizzo `https://github.com/DIUNIPI-ISTI/ConvertCsvToXes`

4.3 Riassunto

Nella prima sezione di questo capitolo è stata descritta la struttura del file di log relativo al processo di Export. Nella seconda sezione sono state descritte le modifiche apportate al plug-in che abbiamo utilizzato per effettuare la trasformazione nel formato XES.

Capitolo 5

ANALISI DEI DATI

In questo capitolo vengono descritte le tecniche di process mining svolte sui log dalla piattaforma TPCS attraverso il framework ProM. In particolare, vengono mostrati e descritti tutti i plug-in che sono stati utilizzati per le analisi, evidenziando i risultati ottenuti. Inizialmente analizzeremo le informazioni che si possono estrarre dal solo log e poi anche quelle ottenibili utilizzando il modello del processo.

5.1 Analisi dei Log

Per poter effettuare le analisi dei log, che avevamo a disposizione, sono stati utilizzati vari plug-in, presenti all'interno di ProM. L'operazione iniziale che deve essere eseguita, prima di fare qualunque tipo di analisi, è quella di importazione del file XES, precedentemente generato, all'interno di ProM. Per fare questo basta, una volta avviato ProM, cliccare sul pulsante "import" e selezionare il file desiderato. Se l'operazione va a buon fine otterremo il risultato mostrato in Figura 5.1.
Nelle sezioni successive verranno descritti nel dettaglio tutti i vari passaggi che sono stati effettuati.

5.1.1 Log Visualizer

La prima analisi è stata effettuata utilizzando il plug-in Log Visualizer. Per poter utilizzare questo plug-in basta importare il file XES all'interno di ProM e selezionare "Log Visualizer" dal combo box posto in alto a destra riportante la scritta "Create new". Appena eseguite queste operazioni si apre una finestra contenente una prima schermata di analisi del log. Vedi Figura 5.2.

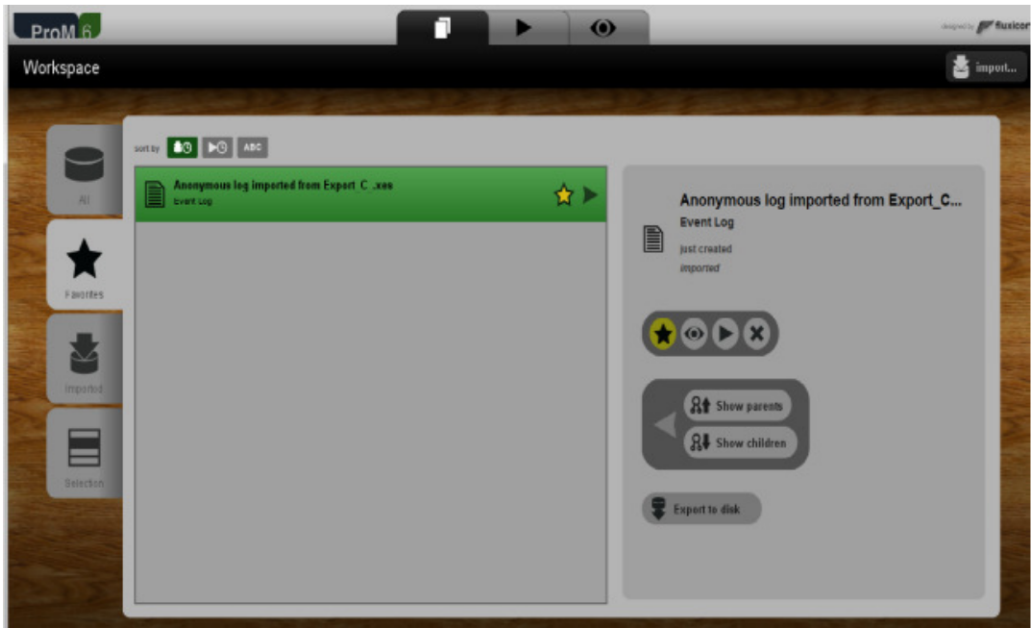

Figura 5.1: Importazione file XES in ProM

Questo plug-in offre una panoramica generale del log, indicando nel dettaglio:

- **Start date:** indica la data del primo evento registrato all'interno del log;
- **End date:** indica la data dell'ultimo evento registrato all'interno del log;
- **Processes:** indica il numero dei processi registrati nel log;
- **Cases:** indica il numero di istanze/tracce presenti all'interno del log;
- **Events:** indica il numero di attività presenti all'interno del log;
- **Event classes:** indica il numero dei tipi di attività presenti all'interno del log;
- **Event type:** indica il numero dei tipi degli eventi, presenti all'interno del log;
- **Originators:** indica il numero di attori che hanno originato le azioni all'interno del log.

Figura 5.2: Vista del plug-in Log Visualizer

Come possiamo vedere dalla Figura 5.2 la data iniziale del log è il 23-05-2014 ore 08:41:58 e quella finale è il 29-05-2014 ore 18:00:28. Nel log sono registrate le attività relative ad un processo e sono presenti 41 **Cases** e 198 **Events**. Gli **Event Classes** sono 8 e gli **Event Types** sono 2 (`START` e `COMPLETE`). Gli **Originators** sono 29. Questo plug-in ci offre anche altre due sezioni contenenti due grafici che mostrano come sono distribuiti gli **Events per case** e gli **Event classes per case**.

Un'altra visualizzazione molto interessante possiamo ottenerla cliccando sul tasto "Inspector", attraverso il quale è possibile visualizzare il dettaglio delle singole tracce. In Figura 5.3 abbiamo selezionato la traccia con il numero di **Events per case** massimo. Come possiamo vedere sono riportati tutti i dati relativi a quella traccia (identificatore della traccia, nome dell'attività svolta, chi ha originato tale attività, data e ora in cui tale attività è stata registrata ed il lifecycle dell'attività).

Infine possiamo cliccare sul pulsante "explorer" per ottenere una visione di insieme delle tracce analizzando il numero di eventi per traccia e la loro frequenza. Come mostrato in Figura 5.4, le istanze del processo sono disposte verticalmente, come flusso di eventi triangolari. Il colore degli eventi descrive la frequenza (verde è molto frequente, rosso è poco frequente). Passando con il mouse sul singolo evento otteniamo maggiori informazioni. Analizzando le varie percentuali possiamo osservare che per ogni attività, la fase `START` e `COMPLETE` hanno la stessa frequenza, quindi possiamo affermare che non ci

Figura 5.3: Inspector del plug-in Log Visualizer

sono mai attività nel log che non si completano. Nel dettaglio, otteniamo i seguenti risultati:

- **APERTURA NAVE:** 95,35%;
- **CONVALIDA NAVE:** 100%;
- **APERTUNA NAVE-VARIAZIONE:** 9,30%;
- **COARRI** 25,58%.

Infine attraverso l'etichetta "Summary" possiamo visualizzare un piccolo sommario del log. Nel nostro caso ci indica che sono presenti 41 Istanze di processo e che il numero totale di eventi è 198.

5.1.2 Show Sequences and Patterns

Attraverso l'utilizzo di questo plug-in è possibile visualizzare tutte le sequenze che sono state generate all'interno del log. Una volta importato il file XES, selezioniamo, dal combo box posto in alto a destra, la voce "Show Sequences and Patterns" ed otteniamo il risultato mostrato in Figura 5.5.

Questo plug-in permette di settare molti parametri in modo da poter affinare l'analisi; possiamo, infatti, selezionare solamente alcune istanze di processo, selezionare solo quelle che sono contenute al di sopra o al di sotto di una certa soglia di tempo, impostare i parametri della sequenza stessa,

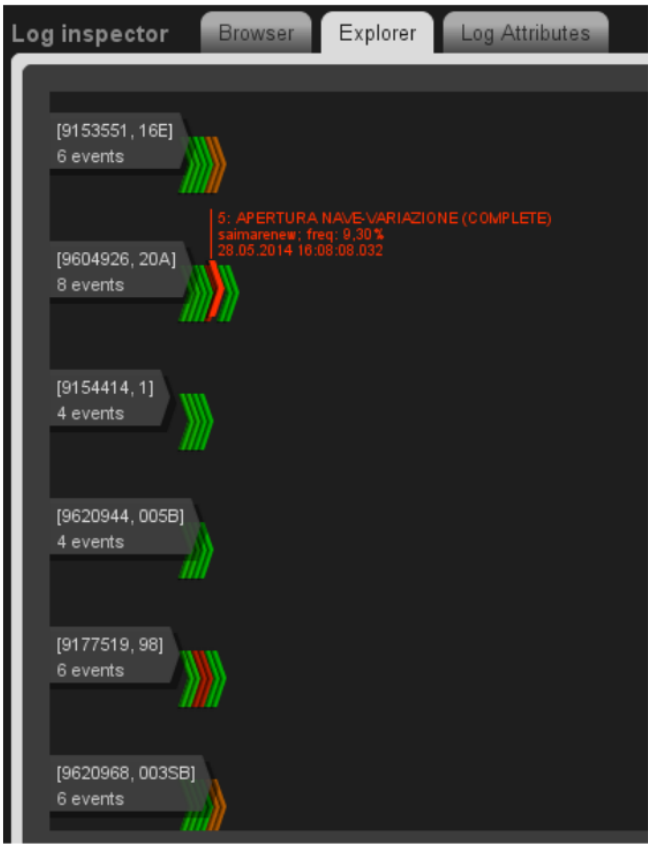

Figura 5.4: Frequenze degli eventi generate attraverso il plug-in Log Visualizer

in modo da considerare come variabile dell'analisi le risorse, l'id, il lifecycle, ecc... Permette inoltre di esportare, in moltissimi formati, il risultato ottenuto.

Per la nostra analisi abbiamo settato la configurazione impostando l'ora come unità di misura del tempo e come componente il nome dell'attività. Come possiamo osservare dalla Figura 5.5 il tempo che intercorre per il passaggio tra la fase di `APERTURA NAVE` e di `CONVALIDA NAVE` è molto breve, mentre il passaggio dalla fase di `CONVALIDA NAVE` a quella di `COARRI` richiede un tempo molto più lungo. Questa caratteristica può essere osservata in tutte le tracce. Inoltre possiamo vedere che il tempo minimo per il verificarsi della fase di `COARRI` è di circa 75 ore rispetto all'inizio del log e che il minor tempo che intercorre per il passaggio dalla fase di `CONVALIDA NAVE` e quella di `COARRI` è di circa 30 ore.

La stessa analisi può essere effettuata andando a ricercare i pattern comu-

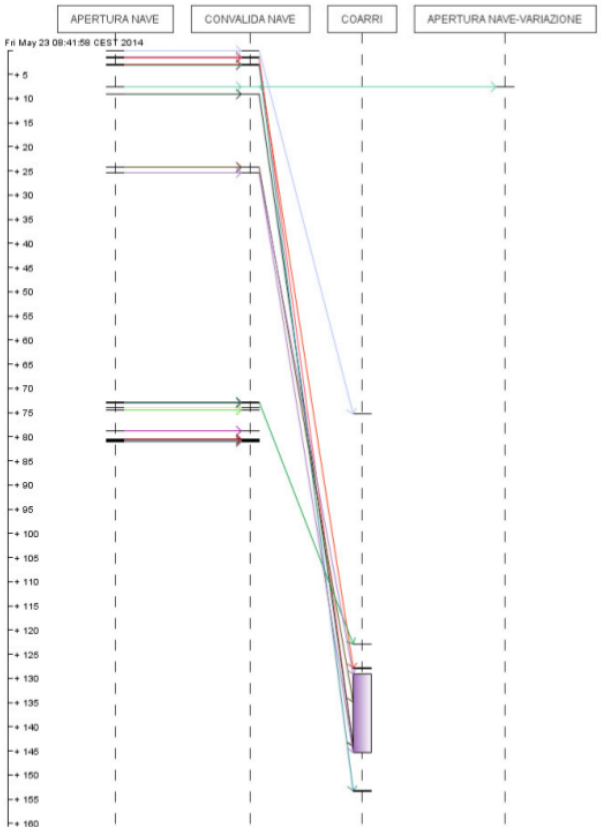

Figura 5.5: Sequenze generate dal Plug-in Show Sequences and Patterns

ni tra le varie tracce. Per fare questo è sufficiente cliccare sul bottone "View patterns" all'interno dell'etichetta di configurazione, ottenendo il risultato riportato in Figura 5.6. Il plug-in ha rilevato la presenza di 5 pattern. Il pattern 0 è caratterizzato dal solo passaggio dalla fase di `APERTURA NAVE` a quella di `CONVALIDA NAVE` e questo passaggio ha un tempo medio di 54 secondi. Il pattern 1 invece aggiunge, rispetto al pattern 0, anche il passaggio da `CONVALIDA NAVE` a `COARRI`. Il tempo medio del primo passaggio tra fasi si alza leggermente, passando a 56 secondi mentre il passaggio da `CONVALIDA NAVE` a `COARRI` ha un tempo medio di 96 ore e mezzo.

I pattern 2 e 3 riguardano i casi in cui si è verificata una variazione nell'apertura della nave che era stata fatta inizialmente, senza il raggiungimento della fase di `COARRI`. Nel pattern 2 è presente solamente un passaggio a ritroso da `APERTURA NAVE-VARIAZIONE` a `CONVALIDA NAVE`, mentre nel pattern 3 è presente un doppio passaggio da `APERTURA NAVE-VARIAZIONE` a `CONVALIDA NAVE` creando la sequenza: `APERTURA NAVE` –> `CONVALIDA NAVE`

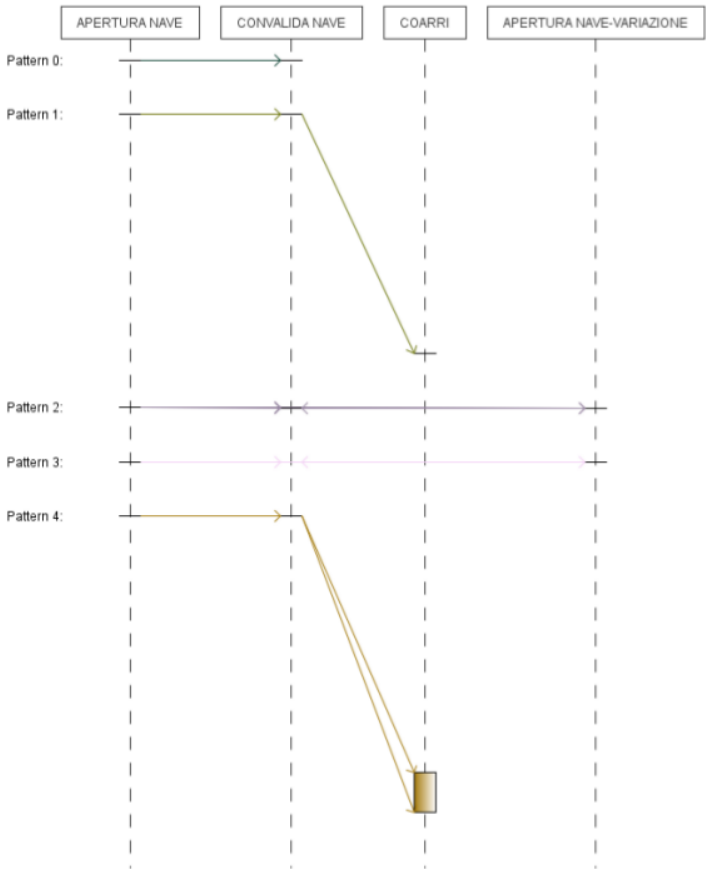

Figura 5.6: Patterns generati dal Plug-in Show Sequences and Patterns

–> `APERTURA NAVE-VARIAZIONE` –> `CONVALIDA NAVE` –> `APERTURA NAVE-VARIAZIONE` –> `CONVALIDA NAVE`. I tempi del pattern 2 risultano quindi inferiori rispetto a quelli del pattern 3, a causa proprio di questo doppio passaggio che si verifica.

Il pattern 4 riguarda una sola traccia che attraversa per due volte la fase di `COARRI` a distanza di 16 ore l'una dall'altra. In merito al passaggio di stato possiamo osservare che il tempo di passaggio dalla fase di `APERTURA NAVE` alla fase `CONVALIDA NAVE` è di circa 30 secondi, mentre il tempo medio che intercorre tra la fase `CONVALIDA NAVE` e la fase `COARRI` è di circa 104 ore.

5.1.3 Synchronous Activity Analysis

Questo plug-in ci permette di analizzare il log dal punto di vista dei tempi che intercorrono per il passaggio da una fase a tutte quelle successive. È in grado di stimare il tempo medio, calcolare il tempo massimo, quello

minimo ed il numero delle occorrenze di ogni singola attività. Una volta

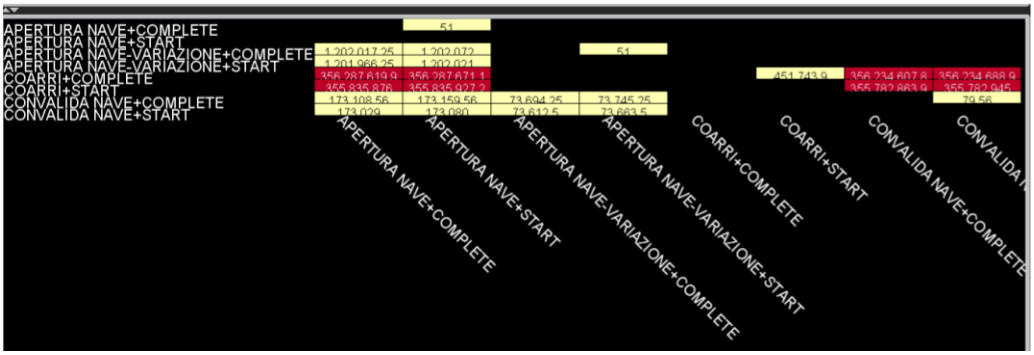

Figura 5.7: Tempi medi passaggio fra stati, generati dal Plug-in Synchronous Activity Analysis

selezionato questo plug-in, ci verrà chiesto come vogliamo classificare i nostri eventi. Sceglieremo "MXML Legacy Classifier" (NomeAttività+Lifecycle), in modo da differenziare la fase di `START` da quella di `COMPLETE` e poi "First Occurrences", in modo da prendere in considerazione la prima occorrenza dell'evento che incontriamo nel log. Come mostrato in Figura 5.7, otteniamo una tabella a doppia entrata con elencati su entrambi gli assi tutte le varie attività presenti nel log. Le caselle vuote stanno ad indicare che non è mai presente il passaggio tra quei due stati. Questo plug-in, non essendo a conoscenza del modello utilizzato, calcola i tempi di passaggio da uno stato all'altro attraverso la differenza dei valori dei timestamp delle due attività prese in considerazione e facendone poi il modulo. Analizzando i dati ottenuti, ci accorgiamo che il tempo che intercorre tra la fase di `START` e quella di `COMPLETE` della stessa attività è molto breve, dell'ordine dei millisecondi, a parte per l'attività di `COARRI`, per la quale intercorrono quasi 8 minuti come tempo medio. Questo risultato potrebbe anche derivare da un errore di registrazione all'interno del log, in quanto il valore massimo è 1 ora e 11 minuti e se lo scartiamo dalla nostra analisi, la media torna ad essere dell'ordine dei millisecondi anche per questa attività.

A fronte di queste considerazioni, possiamo valutare l'ipotesi di considerare le attività, senza tener conto del loro lifecycle. Come possiamo osservare in Figura 5.8, il tempo medio che ci vuole per passare dallo stato di `APERTURA NAVE` a quello di `CONVALIDA NAVE` è di circa 3 minuti, mentre per passare da `APERTURA NAVE` a `APERTURA NAVE-VARIAZIONE` passano in media 20 minuti.

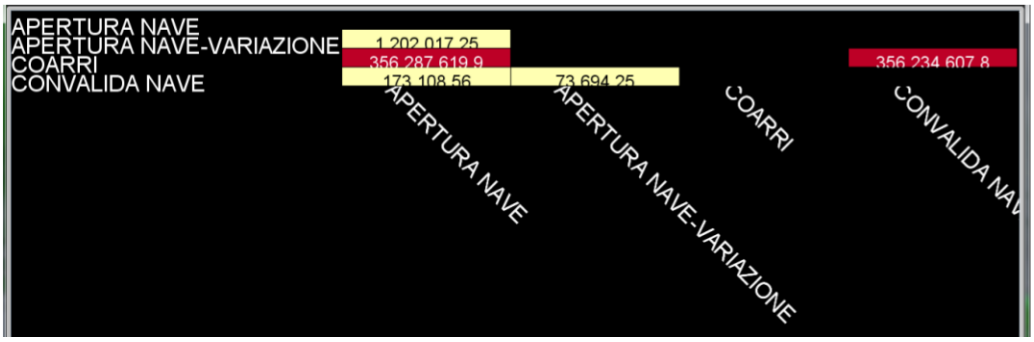

Figura 5.8: Tempi medi passaggio attività, generati dal Plug-in Synchronous Activity Analysis

I tempi si allungano se consideriamo lo stato `COARRI`, infatti ci vogliono più di 4 giorni sia partendo da `APERTURA NAVE` che da `APERTURA NAVE-VARIAZIONE`.

Oltre ai tempi (massimo, minimo e medio) possiamo anche ottenere il numero di occorrenze di passaggi di stato.

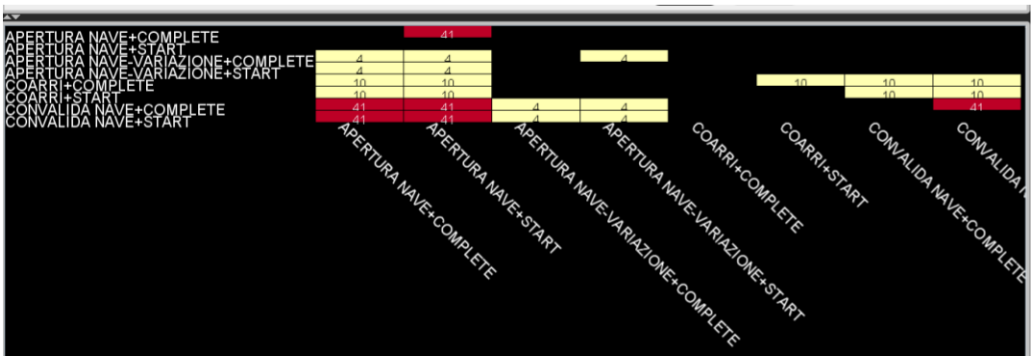

Figura 5.9: Occorrenze delle attività con lifecycle, con Synchronous Activity Analysis

Il risultato ottenuto dall'analisi del nostro log, è riportato in Figura 5.9, e ci mostra che, anche in questo caso, non ci sono differenze tra la fase di `START` e quella di `COMPLETE`, quindi possiamo effettuare la nostra analisi considerando solamente l'attività. La Figura 5.10 mostra che le attività maggiormente presenti all'interno del log sono `CONVALIDA NAVE` e `APERTURA NAVE` con 41 occorrenze, conformemente con i risultati ottenuti dal plug-in "Log

Visualizer". Le attività `COARRI` e `APERTURA NAVE-VARIAZIONE` sono presenti, rispettivamente, 10 e 4 volte.

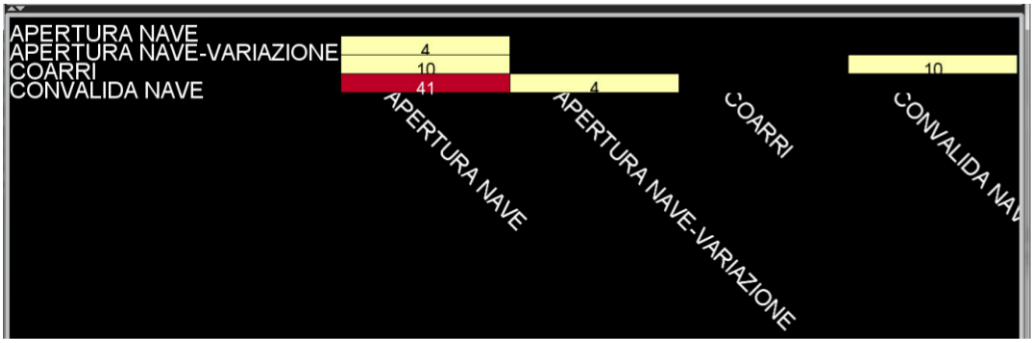

Figura 5.10: Numero di occorrenze delle attività, generate dal Plug-in Synchronous Activity Analysis

5.1.4 Time Based Log Filter

Questo plug-in permette di visualizzare ogni singola traccia collocandola in un arco temporale. Una volta importato il file XES, selezioniamo "Timed Based Log Filter" ed otteniamo il risultato mostrato in Figura 5.11.

Come vediamo in Figura 5.11, sono presenti due sezioni: la sezione di sinistra che permette di settare alcuni parametri e quella di destra che mostra il risultato grafico. Nella nostra analisi abbiamo preferito ordinare le tracce in base al tempo di inizio, questo è stato possibile settando l'opzione "Traces by starting time". Questo plug-in permette anche di selezionare una porzione di queste tracce, in modo da poter effettuare l'analisi solo su un determinato set di tracce. Questo tipo di selezione potrebbe risultare molto utile per analizzare solamente le tracce che iniziano e finiscono all'interno del log, potendo così scartare quelle il cui inizio o la cui fine non sono comprese nel log, oppure settare le tracce con la maggior o minore durata. Una volta selezionate le tracce desiderate, è possibile esportare tale selezione, in modo da poterla usare per elaborazioni future.

5.1.5 Xdotted Chart

Questo plug-in permette di visualizzare le singole attività correlandole con la singola traccia e con il tempo di esecuzione. Una volta importato il file XES,

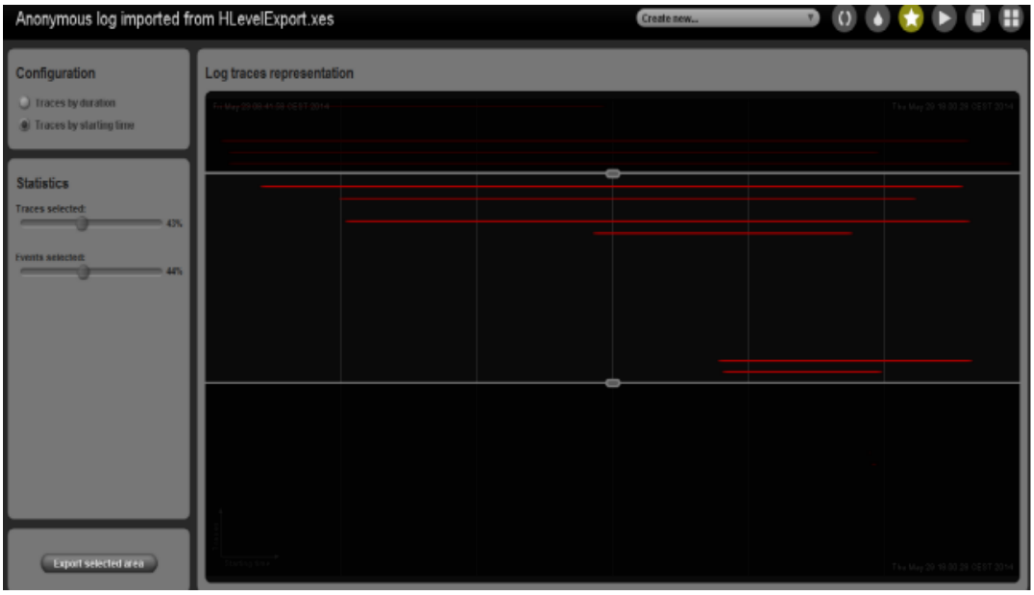

Figura 5.11: Vista del plug-in Time Based Log Filter

selezioniamo “XDotted Chart” ed otteniamo il risultato mostrato in Figura 5.12.

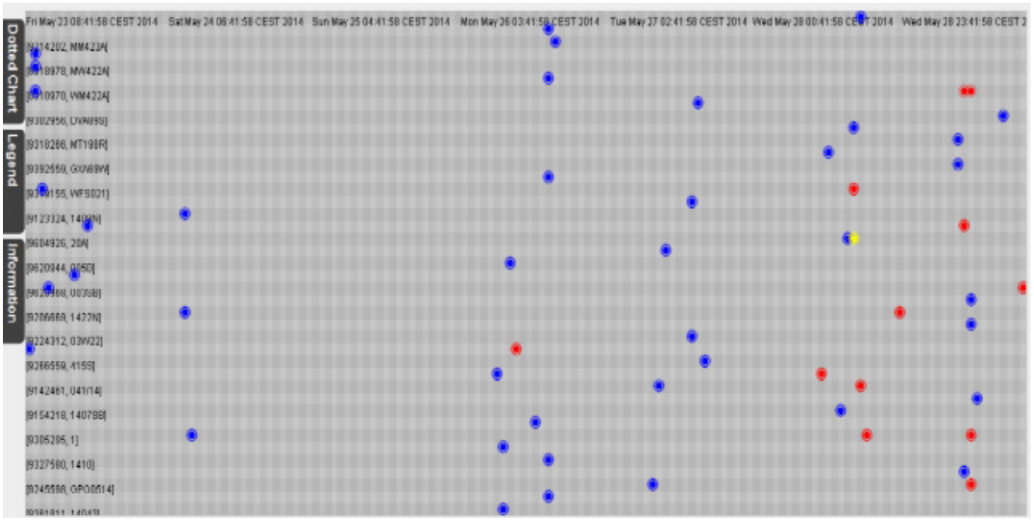

Figura 5.12: Vista del plug-in Xdotted Chart

Anche questo plug-in può essere configurato per eseguire diversi tipi di

analisi. Andando sull'etichetta "Dotted Chart", posta sul lato sinistro dello schermo, compare un menu attraverso il quale è possibile settare i parametri di visualizzazione dei dati contenuti nel log.

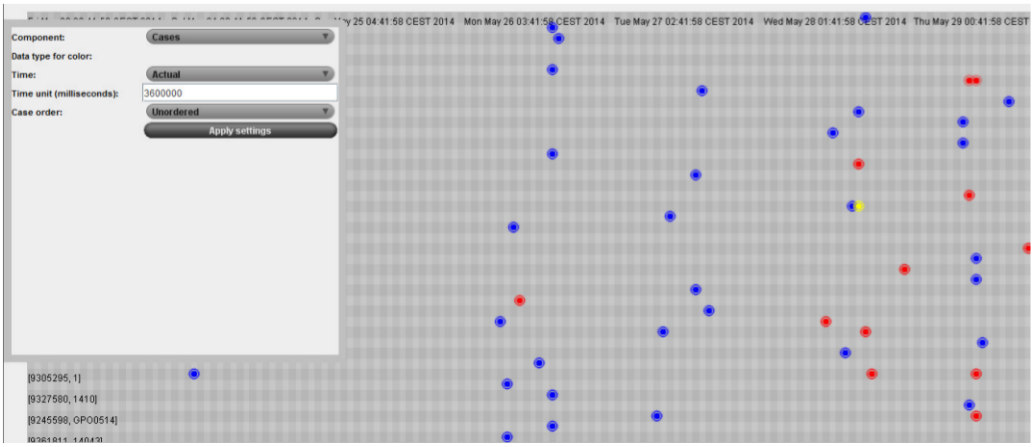

Figura 5.13: Configurazione del plug-in Xdotted Chart

Come mostrato in Figura 5.13 è possibile impostare il componente che desideriamo analizzare (nel nostro caso i "cases"), la tipologia di visualizzazione del tempo (attuale, relativo in base al tempo o ad una percentuale, logico), l'unità di misura del tempo e come ordinare i casi (non ordinati, in base alla durata, in base al numero di occorrenze, in base alla prima o all'ultima occorrenza di un evento). Una volta settati tutti i parametri è sufficiente cliccare sul pulsante "Apply settings" per visualizzare il risultato.

Nel menu posto sul lato sinistro è riportata la legenda, che indica il colore che è stato associato ad ogni tipologia di attività.

L'attività che è maggiormente presente all'interno del log analizzato è `APERTURA NAVE` e possiamo notare che la prima parte del log è caratterizzata esclusivamente da questa attività (in blu), mentre nella seconda parte del log, le attività sono più concentrate e sono presenti alcune fasi di `COARRI` (in rosso).

5.2 Mining dei log

Dopo aver preparato ed analizzato i log, possiamo passare alla fase di Mining dei log, che consiste nel creare un modello partendo da un log.

Per effettuare questo tipo di analisi abbiamo utilizzato alcuni plug-in, presenti all'interno di ProM, che verranno descritti di seguito.

5.2.1 Active Trace Clustering

Questo plug-in, che è stato sviluppato da Jochen De Weerdt [Weerdt et al., 2013], mira alla creazione di cluster di event log per i quali il modello di processo risultante è accurato secondo una precisa metrica. Questo algoritmo, come evidenziato nella Figura 5.14 [Weerdt et al., 2013], si compone di tre fasi distinte: **selection**, **look ahead**, e **residual trace resolution**.

Algorithm 1 ActiTraC

Input: An event log L, the number of clusters nb_{clus}, a target fitness *tf*, a minimum cluster size *mcs*, a window size w, and a boolean N that is true in case a separate cluster should be created for the remaining traces
Output: A collection of event logs, represented by a set of clusters *CS*

Convert L into a grouped event log GL
$CS \leftarrow \emptyset$ # *CS denotes the set of clusters*
$R \leftarrow GL$ # *R denotes the set of remaining dpi's*
while $(|CS| < nb_{clus}) \wedge (R \neq \emptyset)$ **do**
  $C \leftarrow \emptyset$ # *C denotes the set of dpi's in the cluster*
  $I \leftarrow \emptyset$ # *I denotes the set of ignored dpi's*

Phase 1: Selection

repeat
  Define W as the union of the most frequent *dpi* in $R \setminus I$ and the set of the top w% *dpi*'s in $R \setminus I$ according to frequency # *w specifies the size of the window*
  if $C = \emptyset \vee |W| = 1$ **then**
    $cur\_dpi \leftarrow \arg\max_{dpi \in R} |dpi|$
    # *cur_dpi* denotes the *dpi* from R in consideration for addition to the current cluster
  else
    $cur\_dpi \leftarrow \arg\min_{dpi \in W} (dist_{\text{MRA}}(C, dpi))$
    # *function* dist_{MRA} *is the* ***average*** *MRA-based Eucl. distance between the dpi's in C and cur_dpi*
  end if
  $PM \leftarrow HM(C \cup \{cur\_dpi\})$
  if *fitness*$(PM) \geq tf$ **then**
    $C \leftarrow C \cup \{cur\_dpi\}$
    $R \leftarrow R \setminus \{cur\_dpi\}$
  else
    if $\sum_{dpi \in C} |dpi| \geq mcs \times \sum_{dpi \in R} |dpi|$ **then**
      $PM \leftarrow HM(C)$
      Phase 2: Look ahead
      for all $dpi \in R$ **do**
        if *fits*(dpi, PM) **then**
          $C \leftarrow C \cup \{dpi\}$
          $R \leftarrow R \setminus \{dpi\}$
        end if
      end for
      exit repeat
    else
      $I \leftarrow I \cup \{cur\_dpi\}$
    end if
  end if
until $(R = \emptyset) \vee (R = I)$
add the constructed cluster C to *CS*
end while

Phase 3: Residual trace resolution

if N **then**
  add new cluster C to *CS* with $C \leftarrow R$
else
  add each *dpi* in R to that cluster in *CS* for which its fitness, as calculated on the underlying process model, is highest
end if

Figura 5.14: Algoritmo del plug-in Active Trace Clustering

Prima che queste tre fasi possano essere effettuate, le istanze di processo identiche, contenute nell'event log `L`, vengono raggruppate in distinte istanze di processo `dpi`. Inoltre, l'insieme dei clusters creati `CS` viene inizializzato all'insieme vuoto e l'insieme delle distinte istanze di processo rimanenti `R` contiene tutte le `dpi` create nel primo passaggio.

Analizziamo nel dettaglio le tre fasi:

1. **Selezione:** le tracce vengono selezionate in modo iterativo usando una strategia di campionamento selettiva. L'obiettivo è quello di aggiungere una nuova istanza di processo al set di istanze già selezionate, con lo scopo di valutare il modello di processo scoperto da questo nuovo sottoevento di log. Se il modello di processo rimane abbastanza accurato, la traccia selezionata viene aggiunta al cluster corrente e la procedura di selezione viene ripetuta. Il processo continua fino a quando è possibile aggiungere una traccia al cluster corrente senza diminuire troppo

la precisione del modello di processo dando un *target fitness* (`tf`). Il fitness viene utilizzato per controllare se un modello di processo è in grado di riprodurre tutte le sequenze di un log o viceversa, se tutte le tracce di un log sono conformi alla descrizione del modello. Il fitness sarà uguale ad 1 se ogni traccia del log è conforme con la descrizione del modello. L'analisi del fitness ha come scopo quello di rilevare dei disallineamenti tra il modello del processo e l'esecuzione di particolari istanze di process [Rozinat and van der Aalst, 2008].

Per evitare che si creino piccoli gruppi è stato introdotto il parametro *minimum cluster size* (`mcs`) che, variando tra 0 e 1, viene utilizzato per continuare la selezione e la fase di costruzione del modello fino a quando non viene rilevato un caso che si traduce in un modello di processo inadatto (linea 20 dell'algoritmo).

Il parametro *window size* (`w`) in linea 9 dà luogo a due varianti dell'algoritmo ActiTraC, in base al metodo di campionamento. Nel caso in cui `w` sia impostato su 0, la *frequency window* (`W`) conterrà solo il `dpi` più frequente in `R`.

Accanto al campionamento basato sulla frequenza di base, ActiTraC può basarsi su qualsiasi funzione calcolata in base alla distanza tra le istanze di processo distinte. La funzione della distanza alla linea 13 dell'algoritmo viene applicata come segue: dal set corrente di `dpi` residui, una finestra è definita in base alla frequenza. Dal set di `dpi` che formano la finestra, il `dpi` è selezionato in base alla più piccola distanza euclidea media rispetto al set corrente di `dpi` in `C`. In questo modo il cluster risultante sarà significativamente diverso dal primo approccio esclusivamente basato sulla frequenza.

In un modo simile a quello con campionamento selettivo basato sulla frequenza, il `dpi` viene aggiunto al cluster corrente ed un modello di processo (PM) è calcolato con HeuristicsMiner.

Una volta creato questo modello di processo, viene verificato se il fitness del modello di processo è ancora al di sopra di una soglia predefinita, il *target fitness* (`tf`). Se il fitness resta superiore al `tf` , il `dpi` viene aggiunto al set corrente di istanze `C` e viene iniziata una nuova selezione. Tuttavia, se il modello di processo di fitness scende sotto il `tf`, occorre verificare se il minimum cluster size (`mcs`) è raggiunto. Se è così, la selezione viene interrotta ed inizia la seconda fase. Tuttavia, se il `mcs` non è soddisfatto, il `dpi` attualmente selezionato viene aggiunto

all'insieme `I` dei `dpi` saltati e il campionamento selettivo di una nuova traccia da `R` continua.

2. **Look Ahead:** quando viene rilevato che una istanza di processo diminuisce l'accuratezza del modello al di sotto della soglia specificata del cluster corrente e la dimensione minima del cluster viene raggiunta, la prima fase giunge al termine. Nella seconda fase (look ahead) vengono prese in considerazione le istanze rimanenti (ad esempio, quelle tracce che non sono state ancora soggette alla fase di selezione) verificando se alcune di queste tracce sono adatte al modello di processo creato nella prima fase. In questa procedura, solo le istanze di processo distinte, che si adattano perfettamente all'attuale modello di processo (ad esempio, istanze con un fitness uguale a 1), vengono aggiunte al cluster corrente. Le istanze che non si adattano al modello rimangono nell'event log e per esse può essere avviata nuovamente la fase di selezione per creare un secondo cluster. Questa iterazione di selezionare e guardare avanti viene proseguita fino al raggiungimento del numero massimo predefinito di cluster.

3. **Residual Trace Resolution:** la terza fase specifica la risoluzione delle restanti tracce nell'event log. Le istanze rimanenti possono essere separate in un cluster distinto o possono essere distribuite sui cluster creati, secondo il fitness della singola traccia per i diversi modelli di processo creati.

Per eseguire questo plug-in, una volta importato il file XES, basta selezionare il pulsante play e, dopo aver scelto "ActiTraC", premere su "Start". A questo punto si aprono una serie di schermate che consentono la configurazione dei parametri.

Come mostrato in Figura 5.15, possiamo settare molte variabili per configurare il processo di clusterizzazione. Attraverso questa schermata possiamo scegliere il tipo di algoritmo che vogliamo utilizzare, fra i due precedentemente spiegati (noi abbiamo scelto l'algoritmo "Frequency-based selective sampling"), la caratterizzazione del cluster, in particolare la "Minimal cluster size", che serve per definire la dimensione minima di un cluster, e la "Target ICS-fitness". ICS (Improved continuous semantics metric) è una misura del fitness [Bose and van der Aalst, 2010]. Possiamo definire il criterio di stop (noi abbiamo impostato come condizione il numero massimo di cluster a 4) e dove inserire le tracce rimanenti (abbiamo deciso di aggiungerle in un nuovo cluster).

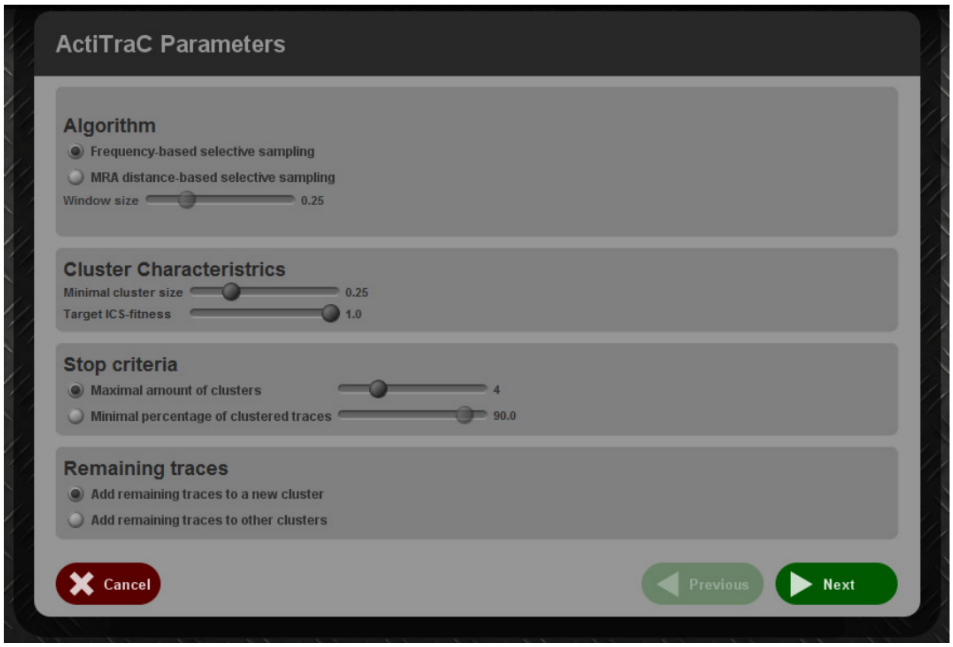

Figura 5.15: Configurazione del plug-in Active Trace Clustering

Dopo aver impostato i parametri, cliccando su "Start" otteniamo il risultato mostrato in Figura 5.16.

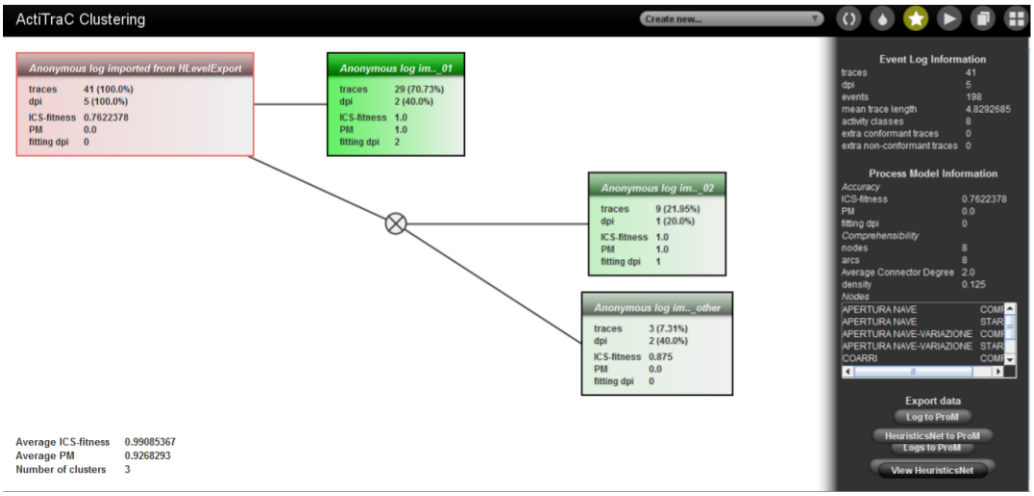

Figura 5.16: Vista del plug-in Active Trace Clustering

Nella nostra analisi, abbiamo ottenuto 3 cluster principali ed un quarto cluster contenente le tracce rimanenti, vedi Figura 5.16, come era stato

settato nelle impostazioni. Cliccando su un cluster possiamo vedere tutte le sue caratteristiche. Il primo cluster contiene tutte le tracce (41) e tutti gli eventi (198). Il fitness di questo cluster è 0,76. Il secondo cluster contiene 29 tracce, circa il 70 percento del totale, e 120 eventi ed il fitness è pari ad 1. Il terzo cluster contiene 9 tracce e 54 eventi ed il fitness è 1 anche in questo caso. Infine abbiamo il cluster con gli elementi residui che contiene 3 tracce e 24 eventi, con un fitness di 0.875.

Possiamo anche visualizzare la visione euristica dei cluster. La Figura 5.17 riporta la vista euristica del primo cluster.

Figura 5.17: Vista euristica del plug-in Active Trace Clustering

Con questo tipo di analisi ci accorgiamo che le tracce contenute nel primo cluster hanno tutte come evento iniziale `APERTURA NAVE+START`. Di queste, 31 terminano la loro esecuzione nella fase di `CONVALIDA NAVE+COMPLETE` e 10 in `COARRI+COMPLETE`.

Anche nel secondo cluster tutte le tracce hanno come evento iniziale `APERTURA NAVE+START`. Di queste 29 tracce, 27 passano dalla fase di `APERTURA NAVE+COMPLETE` alla fase di `CONVALIDA NAVE+START`, mentre due attraversano il ramo che comprende le due fasi di `APERTURA NAVE-VARIAZIONE` (`START` e `COMPLETE`). Tutte le tracce terminano la loro esecuzione nella fase di `CONVALIDA NAVE+COMPLETE`.

Nel terzo cluster tutte le tracce (9) adottano lo stesso comportamento, partendo dalla fase di `APERTURA NAVE+START` passando dalla fase `APERTURA NAVE+COMPLETE`, attraversando la fase di `CONVALIDA NAVE` (`START` e `COMPLETE`) e terminando nella fase di `COARRI+COMPLETE`. Questo terzo cluster è infatti caratterizzato da un *fitness* pari ad 1.

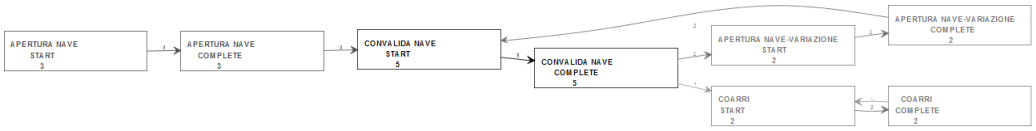

Figura 5.18: Vista euristica del cluster residui con il plug-in Active Trace Clustering

La Figura 5.18 mostra il risultato della vista euristica del cluster contenente gli elementi residui dell'algoritmo sopra spiegato. Questo cluster contiene solo 3 tracce, anch'esse originate nella fase di `APERTURA NAVE+START`. Due di esse si concludono nella fase di `CONVALIDA NAVE+COMPLETE` ed una nella fase di `COARRI+COMPLETE`. Osservando i valori riportati all'interno delle varie fasi, possiamo inoltre vedere che una traccia prevede un doppio passaggio tra le due fasi di `COARRI` (`START` e `COMPLETE`) e che le altre due tracce, dopo la fase di `CONVALIDA NAVE+COMPLETE`, passano entrambe dalle due fasi di `APERTURA NAVE-VARIAZIONE` (`START` e `COMPLETE`), prima di terminare nella fase di `CONVALIDA NAVE+COMPLETE`, che riporta infatti il valore 5.

5.3 Conformance dei log

Dopo aver eseguito le analisi di Mining sulle tracce, siamo passati alla fase di analisi della conformance dei log. Per effettuare tale analisi è necessario avere un modello di processo ed un event log contenente le informazioni che devono essere confrontate. Questo tipo di analisi serve per verificare se quello che è realmente successo (informazioni registrate nell'event log) è conforme al modello e viceversa.

Per effettuare questo tipo di analisi abbiamo utilizzato alcuni plug-in, presenti all'interno di ProM, che verranno descritti di seguito.

5.3.1 Alignment a Log on a Petri Net for Conformance Analisys

Questo plug-in è in grado di accettare una Petri Net ed un event log per creare allineamenti tra ogni traccia nel log e nella rete. Per poter essere eseguito questo plug-in richiede alcune caratteristiche:

1. la rete deve:
   - **a.** avere un marking iniziale non vuoto;
   - **b.** avere un marking finale.
2. il log deve contenere almeno una traccia [Adriansyah, 2012].

Questo plug-in, necessitando in input sia del file contenente gli event log che del modello, è in grado di effettuare una analisi prendendo in considerazione ogni singola traccia e confrontandola col modello. Come primo passo, raggruppa tutti gli eventi che corrispondono ad una traccia ben definita e poi

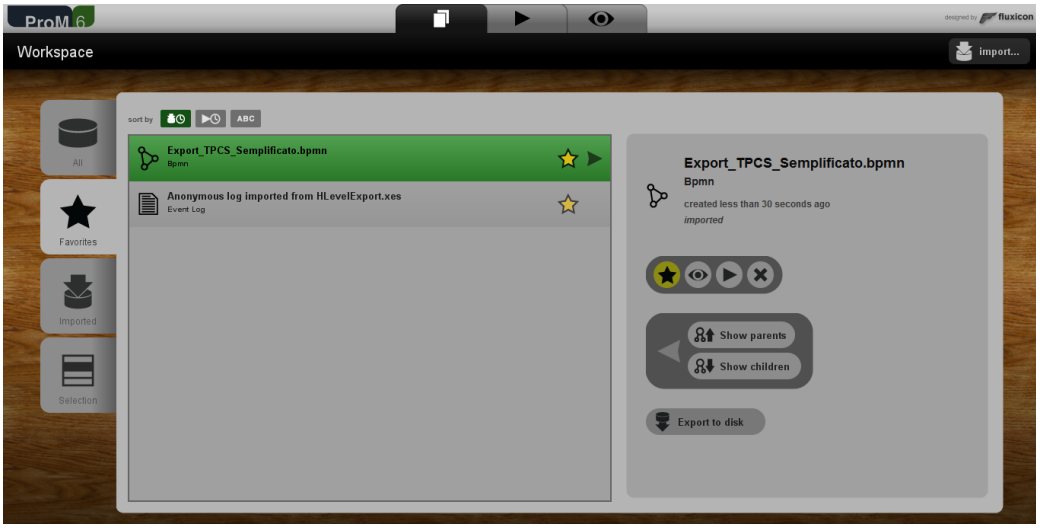

Figura 5.19: Importazione file plug-in Alignment a Log on a Petri Net for Conformance Analisys

li confronta con il modello di processo che è stato trasformato in una rete di Petri. Questo procedimento viene fatto per ogni singola traccia, cercando poi di estrarre comportamenti comuni a più tracce. Attraverso questo confronto dettagliato siamo in grado di capire se un evento, registrato nel log, trova una corrispondenza nel modello, se l'evento è presente solo nel log, se nella traccia che stiamo analizzando manca un passaggio e quindi non può essere associata nessuna registrazione per una determinata fase di una traccia. Inoltre possiamo capire la frequenza con cui si verifica una determinata fase, quali sono gli eventi che originano l'inizio e la fine del processo e quali sono gli errori più comuni che vengono commessi dagli attori. All'interno di questo plug-in ci sono vari modi di analisi e visualizzazione dei risultati ottenuti, in base alla voce selezionata nel menù a tendina presente in alto a destra. Noi ne abbiamo scelti due, che verranno descritti nel seguito.

Fatta questa premessa, andiamo a descrivere tutti i passaggi che abbiamo eseguito per poter avviare questo plug-in. Dopo aver avviato ProM, importiamo sia il file XES (file di log) che il modello semplificato, relativo al processo di Export, come mostrato in Figura 5.19.

A questo punto selezioniamo il file relativo al modello e cliccando sul play si apre la schermata visualizzata in Figura 5.20. Poi selezioniamo il plug-in "Select BPMN Diagram" e clicchiamo su "Start". Questo plug-in serve per settare quale modello vogliamo utilizzare. Nel nostro caso è presente un solo

Figura 5.20: Avvio del plug-in Select BPMN Diagram

modello, come mostrato nella Figura 5.21.

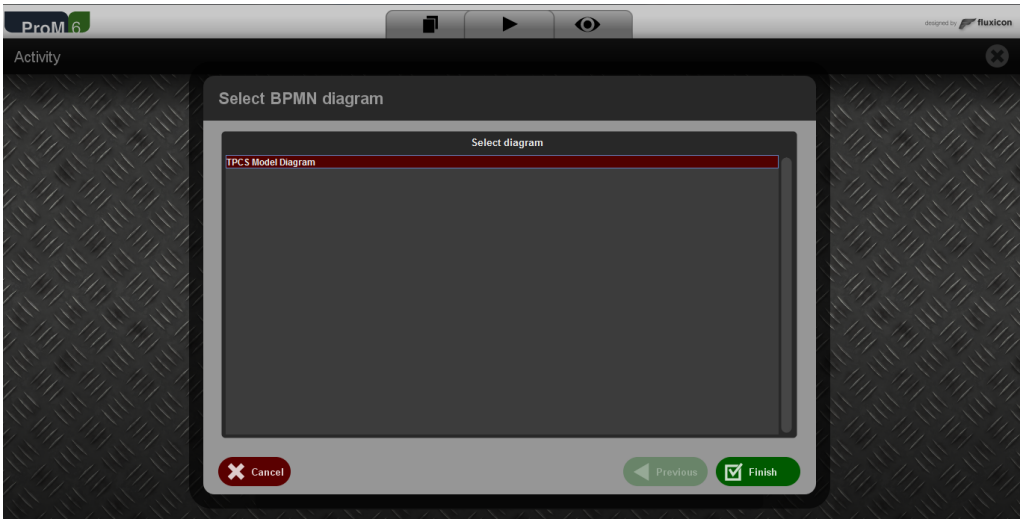

Figura 5.21: Selezione del modello nel plug-in Select BPMN Diagram

Una volta selezionato tale modello clicchiamo sul pulsante “Finish” ed otteniamo il risultato mostrato nella Figura 5.22.

Adesso, per soddisfare i requisiti richiesti dal plug-in che utilizziamo per valutare la conformance, non ci resta che trasformare il modello BPMN ottenuto in una rete di Petri. Una rete di Petri è un grafo orientato bipartito con due tipi di nodi, piazze e transizioni, connessi tra loro da archi. Ogni

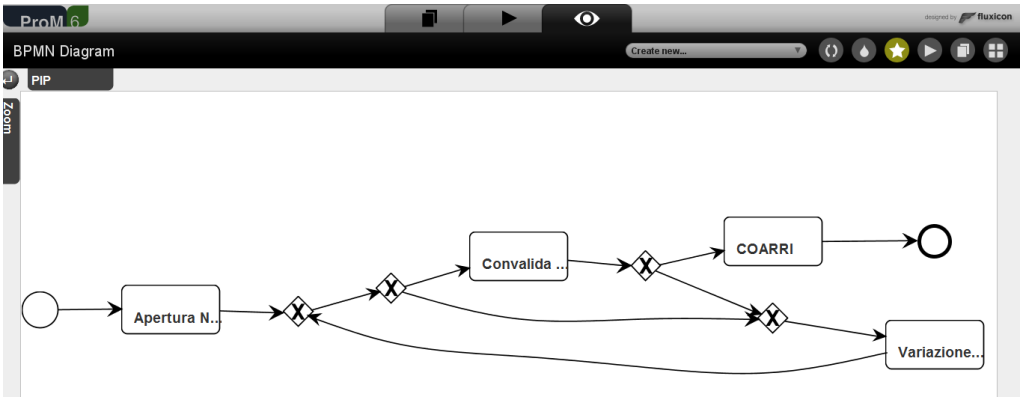

Figura 5.22: Modello ottenuto utilizzando il plug-in Select BPMN Diagram

piazza può essere connessa ad una transizione e viceversa. Due piazze e due transizioni non possono essere connessi tra di loro direttamente. Le piazze sono rappresentate graficamente da cerchi e le transizioni da quadrati. Ogni piazza può contenere al suo interno uno o più token. Il token indica il punto in cui è arrivata l'esecuzione di un dato processo. Una transizione per poter essere eseguita deve avere un token in ogni piazza del suo pre-set.

Per poter generare la corrispondente reti di Petri, utilizziamo il plug-in "BPMN to PetriNet". Per avviare tale plug-in, basta cliccare sul pulsante play del modello BPMN precedentemente generato e selezionarlo nella lista dei plug-in che abbiamo a disposizione, come mostrato in Figura 5.23.

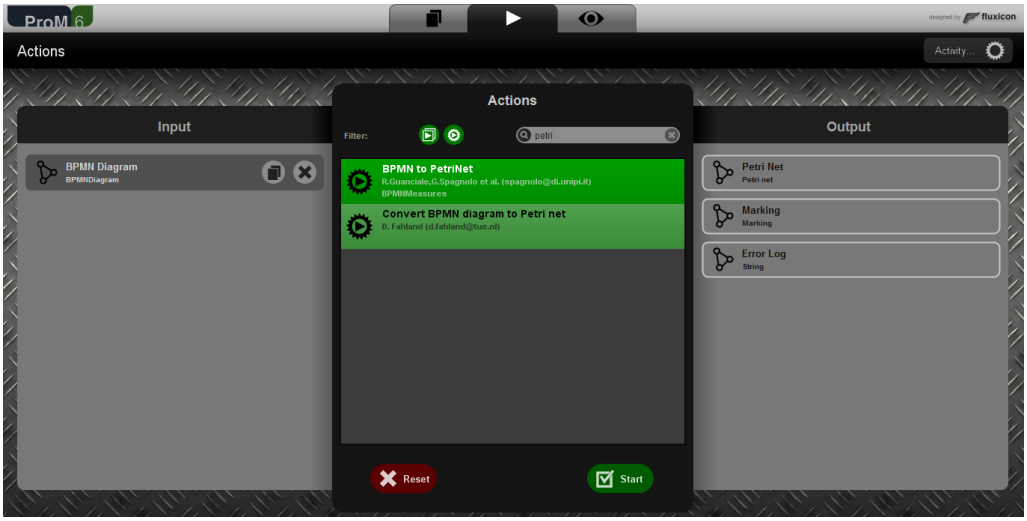

Figura 5.23: Avvio del plug-in BPMN to PetriNet

Premendo il pulsante “Start” otteniamo la rete di Petri rappresentata in Figura 5.24. Osservandola notiamo subito che sono state aggiunte piazze e transizioni per poter rispettare le regole delle reti di Petri; in particolare sono state create le piazze iniziali e finali ed alcune piazze decisionali, attraverso le quali poter indirizzare il flusso correttamente, mantenendo intatte le 8 transizioni che facevano parte del modello iniziale.

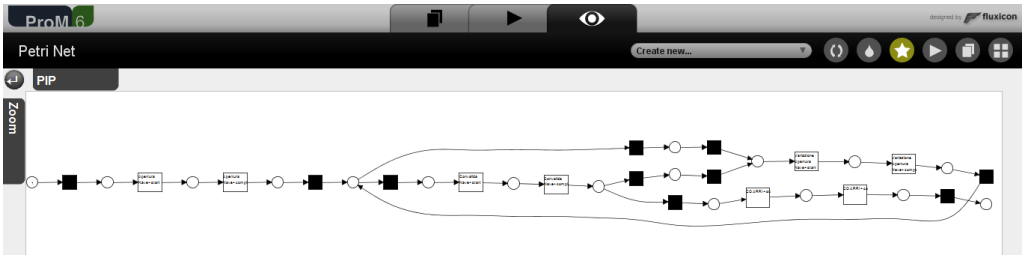

Figura 5.24: Rete di Petri ottenuta utilizzando il plug-in BPMN to PetriNet

Ora siamo veramente in grado di avviare il nostro plug-in per l’analisi della conformance. Dalla rete appena ottenuta, clicchiamo sul pulsante play ed aggiungiamo agli input anche il nostro log iniziale e poi selezioniamo il plug-in “Replay a Log on a Petri Net for Conformance Analisys”, come mostrato in Figura 5.25. Dopo aver cliccato sul pulsante “Start” ci viene chiesto se vogliamo creare un marking finale, in quanto non ne sono stati trovati. Noi clicchiamo su “si” e selezioniamo la piazza “pEnd Event” come marking finale. Se avessimo cliccato su “no” avremmo assunto che tutti i deadlock / stati raggiungibili fossero marking finali, in questo modo però non sarebbe stato possibile eseguire una corretta analisi di conformità.

Il settaggio della configurazione proseguirà con il mapping tra le transizioni trovate all’interno della rete di Petri ed i campi contenuti all’interno del log, come mostrato in Figura 5.26.

Una volta terminato il mapping, cliccando su “Finish”, potremmo settare l’algoritmo che intendiamo usare ed il peso che vogliamo assegnare ad ogni singola transizione. Nella nostra analisi abbiamo lasciato i pesi di default, senza modificare nessun dato. Alla fine otteniamo il risultato mostrato in Figura 5.27, che si riferisce alla vista “Model Projected with Alignments”.

Questa visualizzazione mostra quali task sono spesso ignorati nel modello originale e che sono presenti delle attività extra che non devono essere eseguite secondo il modello ma che vengono in realtà eseguite. Questa vista viene utilizzata per avere una panoramica del momenti in cui si trovano le deviazioni, separando i casi devianti da quelli non-devianti, ed analizzando

Figura 5.25: Avvio del plug-in Replay a Log on a Petri Net for Conformance Analisys

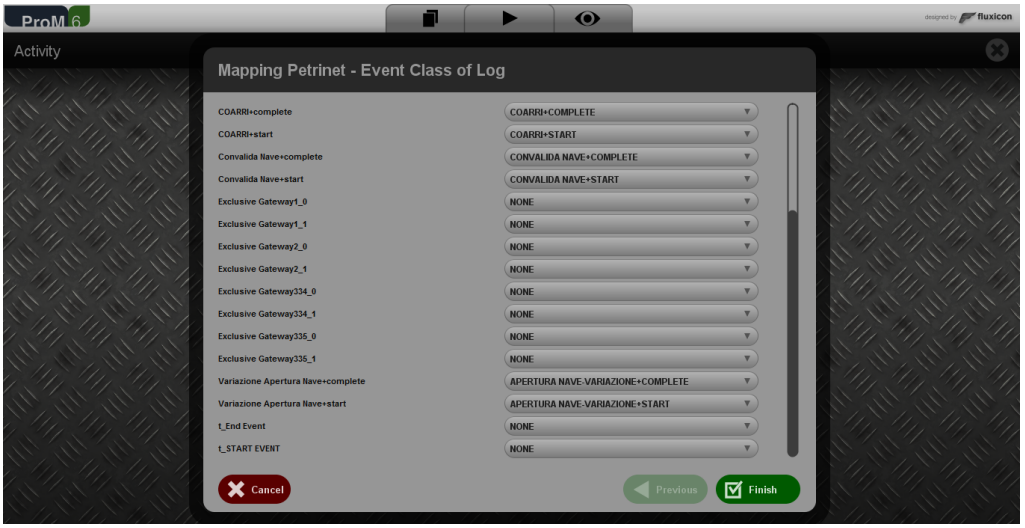

Figura 5.26: Mapping del plug-in Alignment a Log on a Petri Net for Conformance Analisys

le deviazioni che spesso si verificano insieme. Permette inoltre l'esportazione di nuovi log inserendo solamente alcuni case.

La Figura 5.28 mostra le informazioni e le configurazioni che è possibile settare attraverso il pannello "Inspector". All'interno di questo pannello sono presenti informazioni sulla legenda, sulle statistiche degli elementi ed esiste la

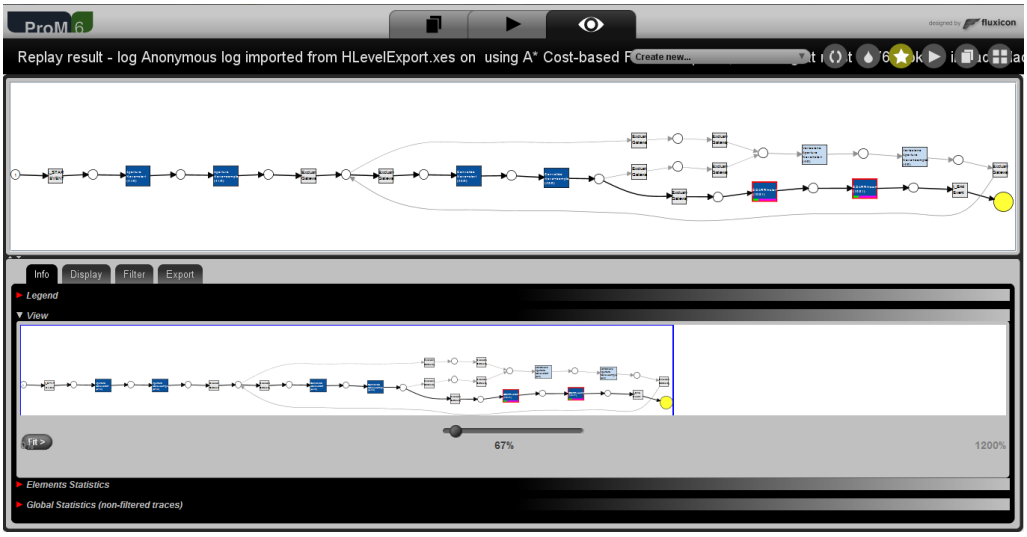

Figura 5.27: Risultato del plug-in Alignment a Log on a Petri Net for Conformance Analisys

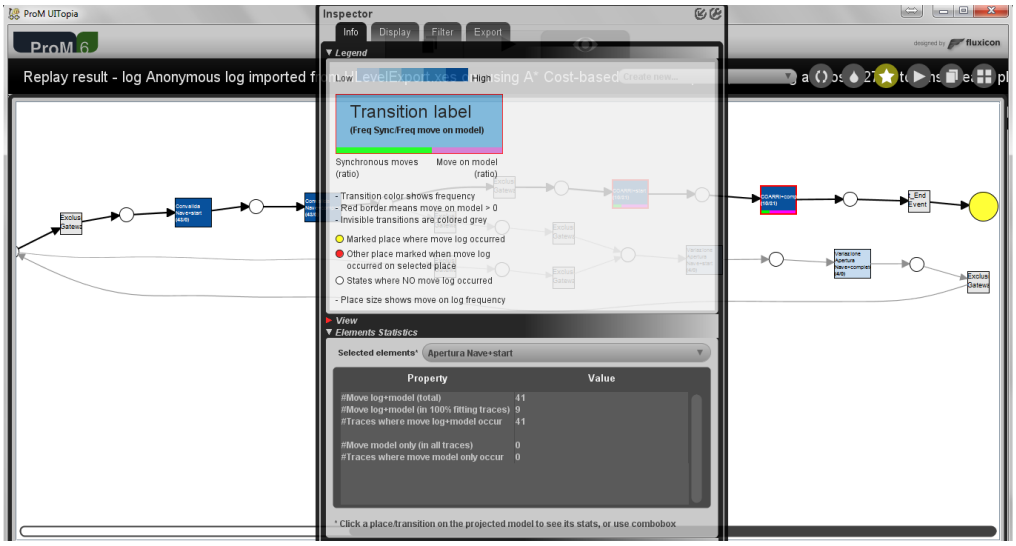

Figura 5.28: Sezione inspector del plug-in Alignment a Log on a Petri Net for Conformance Analisys

possibilità di filtrare i case, ecc... In tale pannello possiamo vedere le statistiche di ogni singola transizione, in merito alla conformità tra il modello ed il log. Osservando il riepilogo ci accorgiamo che le tracce prese in considerazione sono 41 e che di queste 9 hanno un fitting del 100 % nella corrispondenza tra log e modello. Inoltre le corrispondenze di eventi tra il log ed il modello

sono 196 e solamente due attività non hanno corrispondenza tra il log ed il modello.

In alternativa a questo tipo di analisi, possiamo selezionare dal menù in alto a destra "Create new", la voce "Project Alignment to Log" [Adriansyah, 2014]. Il risultato ottenuto è mostrato in Figura 5.29.

Figura 5.29: Risultato della sezione Project Alignment to Log

Questa è la visualizzazione standard degli allineamenti. In questa vista, è possibile ispezionare per ogni traccia se si sono verificati scostamenti, e quali sono le possibili cause. Questo algoritmo ha individuato 5 tipologie principali di allineamenti di tracce.

La prima traccia individuata comprende 27 cases che sono composti, ognuno di essi, da 11 eventi. In questa traccia il log risulta conforme al modello, comprendendo al suo interno solo eventi sincronizzati col modello, oppure presenti solo all'interno del modello (a causa delle piazze e delle transizioni introdotte nella trasformazione in una rete di Petri). Questa tipologia di tracce comprende solo le fasi di `APERTURA NAVE` e di `CONVALIDA NAVE` mentre non è presente la fase di `COARRI`, risultando quindi come saltata nel log.

La seconda traccia è comune a 9 cases ed è anch'essa caratterizzata da 11 eventi. A differenza della prima tipologia di tracce è presente anche la fase di `COARRI`, quindi il log risulta conforme al modello e l'esecuzione risulta completa, partendo dalla piazza iniziale ed arrivando fino a quella finale.

La terza traccia è comune a 2 cases ed è formata da 19 eventi. In questa tipologia di traccia è presente una `APERTURA NAVE-VARIAZIONE`, dopo la fase

di CONVALIDA NAVE e manca la fase di COARRI. Il log risulta conforme al modello, saltando però la fase di COARRI.

La quarta traccia è comune a 2 cases ed è composta da 16 eventi. Rispetto alla precedente tipologia le fasi di CONVALIDA NAVE e APERTURA NAVE-VARIAZIONE sono invertite ed anche in questo caso manca la fase di COARRI, che risulta saltata nel log.

La 5 traccia riguarda un solo case ed è composta da 13 eventi. In questo caso dopo la fase di APERTURA NAVE, seguita da quella di CONVALIDA NAVE, si presenta una prima volta la fase di COARRI, che porterebbe alla piazza finale e quindi all'esecuzione completa del processo. Invece è presente una seconda fase di COARRI che porta ad una non conformità tra il log ed il modello proposto.

Questo caso può essere considerato come un errore commesso dall'operatore ed escludendolo dall'analisi, possiamo affermare che il log risulta conforme al modello fornito.

Al termine di questo progetto di tesi, possiamo affermare che i log analizzati risultano conformi al modello di processo. Questo aspetto risulta fondamentale per l'Autorità Portuale di Livorno, che è particolarmente interessata all'aderenza al modello, in quanto questi risultati possono essere usati per provare tale conformità. Inoltre, confrontando ed unendo i modelli che abbiamo estratto dai log, riusciamo a verificare che non risultano diversi da quello modellato.

5.4 Riassunto

In questo capitolo sono stati descritti tutti i plug-in utilizzati per l'analisi statistica dei log, specificando i vari passaggi eseguiti per effettuare tali analisi e mostrando i risultati ottenuti. Nella seconda parte del capitolo sono state mostrate le tecniche di mining utilizzate, con i relativi risultati. La parte finale del capitolo è stata dedicata all'analisi della conformance dei dati estratti dal log con il modello di processo relativo all'esportazione delle merci.

Capitolo 6

CONCLUSIONI

L'obiettivo di questo progetto di tesi di laurea è stato analizzare un Processo di Business, al fine di ricercare possibili miglioramenti all'interno dello stesso. Questo è stato realizzato attraverso l'utilizzo di tecniche di Process Mining.

Il primo step consisteva nella piena comprensione dei log che ci erano stati forniti e del Business Process associato all'esportazione delle merci. Dopo aver capito la struttura del log, è iniziata la fase di analisi dello stesso. Questa fase è stata caratterizzata dall'utilizzo massivo di ProM, un tool di Process Mining che fornisce strumenti per questo tipo di analisi. Questo lavoro è utile per individuare eventuali problemi all'interno del processo che potrebbero essere causati da errori da parte degli attori, da errori logici/procedurali commessi nello sviluppo del modello di processo o solo da ritardi, dovuti ad esempio a colli di bottiglia. La varietà di plug-in presenti all'interno di ProM ci ha permesso di analizzare i log da vari punti di vista e poter effettuare un'approfondita analisi di conformance tra i log ed il processo. Tale analisi poteva essere maggiormente approfondita, in quanto non avevamo a disposizione una grande mole di dati, che nello specifico riguardava solamente una settimana di registrazione delle operazioni. Se avessimo avuto un periodo più grande da analizzare, la qualità e la significatività dei risultati ottenuti sarebbe potuta essere maggiore.

Questo progetto di tesi ha permesso di approfondire maggiormente le conoscenze sui Business Process, sul Process Mining ed in particolare sulle varie tecniche che possono essere utilizzate per eseguire queste analisi. Anche dal punto di vista pratico, l'utilizzo del tool ProM, ha permesso di approcciarsi direttamente ad un problema in un contesto reale come quello del Porto di Livorno. Molto utile è stata la fase di selezione dei plug-in all'interno di ProM, al fine di trovare quelli che si adattassero maggiormente al tipo di analisi richiesta, che ha portato allo studio ed alla conoscenza di ulteriori

plug-in oltre a quelli descritti in questo progetto di tesi.

Un altro aspetto che è stato approfondito è quello riguardante gli event log ed il formato XES, in quanto considerato come il formato standard utilizzato all'interno di ProM.

Al termine della fase di studio e delle varie analisi effettuate possiamo concludere che tutte le aziende dovrebbe implementare e dare maggiore importanza ai Business Process, in quanto avere processi aziendali ben definiti serve sicuramente sia per l'utilizzatore, il quale è facilitato nel seguire una linea guida prestabilita, che per l'azienda che mira a massimizzare la sua produttività. I Business Process dovrebbero essere integrati con delle registrazioni dettagliate di tutte le operazioni che vengono effettuate, in modo da poter essere confrontati con il comportamento vero e proprio che viene seguito. Per questo è fondamentale dedicare sempre maggior attenzione alla creazione di event log contenenti tutte le informazioni necessarie. Sarebbe importante inserire all'interno del log tutte le tipologie di informazioni che potrebbero risultare necessarie, senza esagerare, inserendo dati totalmente inutili, per non rischiare di far aumentare in maniera esponenziale la dimensione dei log.

L'obiettivo delle aziende dovrebbe essere quello di far diventare i processi di business e la registrazione delle azioni svolte negli event log, come parte integrante della produttività. Anche se l'inserimento iniziale di questi processi potrebbe trovare una forte opposizione da parte degli attori, magari abituati da anni ad agire in un certo modo, alla fine verrebbero gradualmente accettati grazie alla facilità di lettura, interpretazione ed utilizzo e porterebbero un netto miglioramento nelle performance aziendali. Inoltre, grazie all'azione di controllo e monitoraggio, sarebbe più facile riscontrare abitudini errate, motivi dei ritardi o negligenze da parte degli attori.

Bibliografia

[Adriansyah, 2012] Adriansyah, A. (2012). *Replay a Log on Petri Net for Conformance Analysis Plug-in*. `http://goo.gl/88Jhlh`.

[Adriansyah, 2014] Adriansyah, A. (2014). *Aligning observed and modeled behavior*. SIKS dissertatiereeks; 2014-07. Eindhoven: Technische Universiteit Eindhoven.

[Bose and van der Aalst, 2010] Bose, R. and van der Aalst, W. (2010). *Trace Clustering Based on Conserved Patterns: Towards Achieving Better Process Models*. In Rinderle-Ma, S., Sadiq, S., and Leymann, F., editors, *BPM 2009 Workshops, Proceedings of the Fifth Workshop on Business Process Intelligence (BPI'09)*, volume 43 of *Lecture Notes in Business Information Processing*, pages 170–181. Springer-Verlag, Berlin.

[Günther and Verbeek, 2014] Günther, C. W. and Verbeek, E. (2014). *XES - Standard Definition*. `http://www.xes-standard.org/_media/xes/xesstandarddefinition-2.0.pdf`.

[OMG, 2011] OMG (2011). BPMN 2.0. `http://www.bpmn.org/`.

[Rozinat and van der Aalst, 2008] Rozinat, A. and van der Aalst, W. (2008). *Conformance checking of processes based on monitoring real behavior. Inf. Syst.*, 33(1):64–95.

[van der Aalst and Adriansyah, 2012] van der Aalst, W. and Adriansyah, A. (2012). *Process mining manifesto*. In Daniel, F., Barkaoui, K., and Dustdar, S., editors, *Lecture Notes in Business Information Processing*, volume 99, pages 169–194. Springer.

[Verbeek and Günther, 2014] Verbeek, E. and Günther, C. W. (2014). *ProM 6*. `http://www.promtools.org/prom6/`.

[Verbeek, 2010a] Verbeek, H. E. (2010a). *ProM 6 Getting Started*. http://www.promtools.org/prom6/downloads/prom-6.0-getting-started.pdf.

[Verbeek, 2010b] Verbeek, H. E. (2010b). *ProM 6 Tutorial*. http://www.promtools.org/prom6/downloads/prom-6.0-tutorial.pdf.

[Weerdt et al., 2013] Weerdt, J. D., vanden Broucke, S. K. L. M., Vanthienen, J., and Baesens, B. (2013). *Active Trace Clustering for Improved Process Discovery. IEEE Trans. Knowl. Data Eng.*, 25(12):2708–2720.